NUTAN MAHARASHTRA VIDYA PRASARAK MANDAL'S

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING AND TECHNOLOGY

AN AUTONOMOUS INSTITUTE | UNDER ADMINISTRATIVE SUPPORT OF PCET

Curriculum Structure and Syllabus of M. Tech. Computer Engineering (2026 Pattern)

VISION OF THE INSTITUTE

To be a notable institution for providing quality technical education and ensuring ethical, moral and holistic development of students.

MISSION OF THE INSTITUTE

To nurture engineering graduates with state of the art competence, professionalism and problem solving skills to serve needs of industry as well as society.

VISION OF COMPUTER ENGINEERING

Imbibing Quality Technical Education and Overall Development by Endowing Students with Societal and Ethical skills in Computer Engineers.

MISSION OF COMPUTER ENGINEERING

- To impart engineering knowledge and skills by adopting effective teaching learning processes.
- To develop professional, entrepreneurial & research competencies encompassing continuous intellectual growth.
- To produce educated students to exhibit societal and ethical responsibilities in the working environment.

INDEX

COURSE-WISE CREDIT DISTRIBUTION

Sl. No.	Type of Course	No. of Courses	Total Credits	
			No.	%
1.	Programme Core Course (PCC)	7	24	27
2.	Programme Elective Course (PEC)	4	11	13
3.	Open Elective (OE)	1	3	3
4.	Experiential Learning (Research Methodology) Course	4	14	16
5.	Experiential Learning (Internship/ OJT) Course	1	5	6
6.	Experiential Learning (Project) Course	2	27	31
7.	Self-Learning Course	1	4	4
TOTAL		**20**	**88**	**100**

SEMESTER-WISE COURSE DISTRIBUTION

Sr. No.	TYPE OF COURSE	NO. OF COURSES / SEMESTER				TOTAL
		1	2	3	4	
1.	Programme Core Course (PCC)	3	4	-	-	7
2.	Programme Elective Course (PEC)	3	1	-	-	4
3.	Open Elective (OE)	-	1	-	-	1
4.	Experiential Learning (Research Methodology) Course (ELC)	1	1	1	1	4
5.	Experiential Learning (Internship/ OJT) Course	-	-	1	-	1
6.	Experiential Learning (Project) Course	-	-	1	1	2
7.	Self-Learning Course (SLC)	-	-	1	-	1
Total		**7**	**7**	**4**	**2**	**20**

CURRICULUM STRUCTURE
M. Tech. Computer Engineering

Semester – I

First Year M. Tech Computer Engineering																	
Semester I																	
Sl. No.	**Course Code**	**Course Type**	**Course Name**	**Credits**			**Teaching Scheme (Hours/Week)**			**Examination Scheme and Marks**							
										CCE		**ESE**					
				TH	**TUT**	**PR**	**L**	**T**	**P**	**UT**	**FA**	**SA**	**PR**	**OR**	**TW**	**TOTAL**	
										25	**25**	**50**					
1	MCE26PCC-501	Programme Core Course	Advanced Algorithms	4			4			25	25	50				**100**	
2	MCE26PCC-502	Programme Core Course	Advanced Machine Learning	4			4			25	25	50				**100**	
3	MCE26PCC-503	Programme Core Course	Computational Laboratory – I			2			4					25	25	**50**	
4	MCE26PEC-504	Programme Elective Course	Programme Elective Course - I	3			3			25	25	50				**100**	
5	MCE26PEC-505	Programme Elective Course	Programme Elective Course-II	3			3			25	25	50				**100**	
6	MCE26PEC-506	Programme Elective Course	Project Based Learning			2			4					25	25	**50**	
7	MCE26ELC - 507	Experiential Learning (Research Methodology) Course	Research Methodology	4			4			25	25	50				**100**	
TOTAL				**18**		**4**	**18**		**8**	**125**	**125**	**250**		**50**	**50**	**600**	
				22			**26**										

CCE- Comprehensive Continuous Evaluation**, ESE**- End Semester Evaluation, **TW**-Term Work, **OR**-Oral, **PR**-Practical, **TH**- Theory**, L**-Lecture, **T/TUT**-Tutorial, **UT**- Unit Test, **FA**–Formative Assessment, **SA** – Summative Assessment

Basket: List of Courses – Program Elective Course-I

Course Code	Course Name	
MCE26PEC-504A	Data Engineering	Choose Any One
MCE26PEC-504B	Computer Vision	
MCE26PEC-504C	Advanced Computer Networks	

Basket: List of Courses – Program Elective Course-II

Course Code	Course Name	
MCE26PEC-505A	Cloud Computing	Choose Any One
MCE26PEC-505B	Cryptography and Cryptanalysis	
MCE26PEC-505C	Deep Learning	

CURRICULUM STRUCTURE
M. Tech. Computer Engineering

Semester – II

First Year M. Tech Computer Engineering																	
Semester II																	
Sl. No.	**Course Code**	**Course Type**	**Course Name**	**Credits**			**Teaching Scheme (Hours/Week)**			**Examination Scheme and Marks**							
										CCE		**ESE**					
				TH	**TUT**	**PR**	**L**	**T**	**P**	**UT**	**FA**	**SA**	**PR**	**OR**	**TW**	**TOTAL**
										25	**25**	**50**				
1	MCE26PCC-551	Programme Core Course	Advanced Databases	4			4			25	25	50				**100**
2	MCE26PCC-552	Programme Core Course	Quantum Computing	4			4			25	25	50				**100**
3	MCE26PCC-553	Programme Core Course	Natural Language Processing	4			4			25	25	50				**100**
4	MCE26PCC-554	Programme Core Course	Computational Laboratory – II			2			4					25	25	**50**
5	MCE26PEC-555	Programme Elective Course	Programme Elective Course-III	3			3			25	25	50				**100**
6	-	Open Elective	Open Elective	3			3			25	25	50				**100**
7	MCE26ELC-557	Experiential Learning (Research Methodology) Course	Research Seminar-I			2			4					25	25	**50**
TOTAL				**18**		**4**	**18**		**8**	**125**	**125**	**250**		**50**	**50**	**600**
				22			**26**									

CCE- Comprehensive Continuous Evaluation**, ESE**- End Semester Evaluation, **TW**-Term Work, **OR**-Oral, **PR**-Practical, **TH**- Theory**, L**-Lecture, **T/TUT**-Tutorial, **UT**- Unit Test, **FA**–Formative Assessment, **SA** – Summative Assessment

Basket: List of Courses – Program Elective Course-III

Course Code	Course Name	
MCE26PEC-555A	DevOps Engineering	Choose Any One
MCE26PEC-555B	Block chain Technology	
MCE26PEC-555C	Generative Artificial Intelligence	

Basket: List of Courses – Open Elective

Course Code	Course Name	
MCE26OEC-556A	Strategic Management And Business Analytics	Choose Any One
MCE26OEC-556B	Software Project Management	
MET26OEC-556C	Cost Management Of Engineering Projects	

CURRICULUM STRUCTURE
M. Tech. Computer Engineering

Semester – III

Second Year M. Tech Computer Engineering																	
Semester III																	
Sl. No.	**Course Code**	**Course Type**	**Course Name**	**Credits**			**Teaching Scheme (Hours/Week)**			**Examination Scheme and Marks**							
										CCE		**ESE**					
				TH	**TUT**	**PR**	**L**	**T**	**P**	**UT**	**FA**	**SA**	**PR**	**OR**	**TW**	**TOTAL**	
										25	**25**	**50**					
1	MCE26SL-601	Self-Learning Course	Massive Open Online Course	4											100	**100**	
2	MCE26ELC-602	Experiential Learning (Internship/ OJT) Course	Internship			5			10						100	**100**	
3	MCE26ELC-603	Experiential Learning (Research Methodology) Course	Research Seminar-II			4			8					25	25	**50**	
4	MCE26ELC -604	Experiential Learning (Project) Course	Research Project – I			9			18					25	25	**50**	
TOTAL				**4**		**18**			**36**					**50**	**250**	**300**	
				22			**36**										

CCE- Comprehensive Continuous Evaluation**, ESE-** End Semester Evaluation, **TW**-Term Work, **OR**-Oral, **PR**-Practical, **TH**- Theory**, L**-Lecture, **T/TUT**-Tutorial, **UT**- Unit Test, **FA**–Formative Assessment, **SA** – Summative Assessment

CURRICULUM STRUCTURE
M. Tech. Computer Engineering

Semester – IV

Second Year M. Tech Computer Engineering																	
Semester IV																	
Sl. No.	**Course Code**	**Course Type**	**Course Name**	**Credits**			**Teaching Scheme (Hours/Week)**			**Examination Scheme and Marks**							
										CCE		**ESE**					
				TH	**TUT**	**PR**	**L**	**T**	**P**	**UT**	**FA**	**SA**	**PR**	**OR**	**TW**	**TOTAL**	
										25	**25**	**50**					
1	MCE26ELC - 651	Experiential Learning (Research Methodology) Course	Research Seminar III			4			8					50	50	**100**	
2	MCE26ELC - 652	Experiential Learning (Project) Course	Research Project – II			18			36					50	150	**200**	
TOTAL						**22**			**36**					**100**	**200**	**300**	
				22			**44**										

CCE- Comprehensive Continuous Evaluation**, ESE**- End Semester Evaluation, **TW**-Term Work, **OR**-Oral, **PR**-Practical, **TH**- Theory**, L**-Lecture, **T/TUT**-Tutorial, **UT**- Unit Test, **FA**–Formative Assessment, **SA** – Summative Assessmet

Course Syllabus

Semester-I

Program	F.Y. M. Tech (Computer Engineering)			Semester: I			
Course	Advanced Algorithms			Code:	MCE26PCC-501		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
4	4	-	-	25	25	50	100

Pre-requisites: Prior knowledge of foundation in Data Structures and Algorithms is required.

Course Objectives: This course aims at enabling students:

1. To understand advanced algorithm design techniques.
2. To analyze performance of algorithms in terms of time and space complexity.
3. To apply algorithmic strategies for solving computational problems.
4. To understand randomized, multithreaded, and distributed algorithms.
5. To study optimization techniques and computational complexity.

Course Outcome: After completion of the course, the students will be able to:

CO1: Evaluate the efficiency of algorithms using worst, best, and average case analysis.

CO2: Apply dynamic programming techniques to solve problems such as binomial coefficients, matrix chain multiplication, and longest common subsequence

CO3: Design solutions using divide and conquer, greedy, and backtracking strategies.

CO4: Analyze the role of randomized and parallel algorithms in modern computing environments.

CO5: Classify problems into P, NP, NP-complete categories and apply reduction techniques.

Course Contents

Unit	Description	Duration [Hrs]
I	**Foundations of Algorithm Analysis** Role of algorithms in computing. Growth of functions and asymptotic notations (O, Ω, Θ). Recurrence relations and methods of solving recurrences including substitution method, recursion tree method and Master theorem. Worst-case, best-case and average-case analysis. Review of algorithmic strategies such as greedy method, divide and conquer, dynamic programming and branch and bound method.	12
II	**Dynamic Programming and Linear Programming** Principles and control abstraction of dynamic programming. Applications of dynamic programming including binomial coefficients, matrix chain multiplication and longest common subsequence. Introduction to linear programming, formulation of problems as linear programming models, simplex method and concept of duality.	12
III	**Randomized, Parallel and Distributed Algorithms** Concept and need of randomized algorithms. Randomized Quick Sort and basic idea of min-cut algorithm. Introduction to approximation algorithms with Travelling Salesman Problem example. Parallel algorithms including parallel loops, race conditions, multithreaded	12

	merge sort and multithreaded matrix multiplication (conceptual understanding). Introduction to distributed algorithms including distributed breadth first search and distributed minimum spanning tree. String matching algorithms including naive approach and Rabin-Karp algorithm.	
IV	**Geometric and Optimization Algorithms** Introduction to computational geometry. Algorithms for convex hull, closest pair of points, range searching and segment intersection. Introduction to optimization algorithms including gradient descent, genetic algorithms and particle swarm optimization. Applications in logistics, clustering and decision-making systems.	**12**
V	**Complexity Theory** Lower bound concepts. Complexity classes P, NP, NP-complete and NP-hard. Polynomial time reductions and Cook's theorem (conceptual understanding). Examples including SAT, Travelling Salesman Problem, Hamiltonian cycle and subset sum. Discussion on P versus NP and its practical implications.	**12**
	Total	**60**

Text Books:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., Introduction to Algorithms, 4th Edition, MIT Press, 2022. ISBN: 978-0262046305.
2. Dave, P. H., and Dave, H. B., Design and Analysis of Algorithms, 2nd Edition, Pearson Education, 2013. ISBN: 978-8131760826.

Reference Books:

1. Motwani, R., and Raghavan, P., Randomized Algorithms, Cambridge University Press, 2004. ISBN: 978-0521613903.
2. Brassard, G., and Bratley, P., Fundamentals of Algorithmics, Prentice Hall of India (PHI), 1996. ISBN: 978-8120311312.
3. Goodrich, M. T., and Tamassia, R., Algorithm Design: Foundations, Analysis and Internet Examples, Wiley, 2006. ISBN: 978-8126509867.
4. Horowitz, E., Sahni, S., and Mehta, D., Fundamentals of Computer Algorithms, 2nd Edition, Universities Press, 2008. ISBN: 978-8173716126.

e-Books:

1. https://www.deeplearningbook.org

MOOC / NPTEL/YouTube Links:

1. Design and Analysis of Algorithms, Prof. Madhavan Mukund, Chennai Mathematical Institute, SWAYAM-NPTEL Course, Available at: https://onlinecourses.nptel.ac.in/noc19_cs47/preview
2. Advanced Algorithms, Prof. Naveen Garg, Indian Institute of Technology Delhi, SWAYAM-NPTEL Course, Available at: https://onlinecourses.nptel.ac.in/noc20_cs88/preview
3. Design and Analysis of Algorithms, Prof. Abhiram Ranade, Indian Institute of Technology Bombay, NPTEL Course, Available at: https://nptel.ac.in/courses/106101060

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Advanced Machine Learning			Code:			MCE26PCC-502
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
4	4	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Python Programming, and basic Machine Learning concepts is required.

Course Objectives: This course aims at enabling students:

1. To understand advanced supervised learning techniques for predictive modeling.
2. To learn unsupervised and representation learning methods for pattern discovery.
3. To explore deep learning architectures such as CNNs, RNNs, and Transformers.
4. To understand advanced machine learning concepts including Reinforcement Learning and generative models.
5. To learn model deployment, MLOps practices, and ethical AI principles.

Course Outcome: After completion of the course, the students will be able to:

CO1: Apply advanced ML algorithms to complex real-world problems.
CO2: Design deep learning architectures for structured and unstructured data.
CO3: Implement modern ML systems including Transformers and LLMs.
CO4: Deploy scalable ML models using MLOps practices.
CO5: Analyze research papers and develop innovative ML solutions.

Course Contents

Unit	Description	Duration [Hrs]
I	**Foundations and Advanced Supervised Learning** Review of Machine Learning fundamentals, Bias-Variance tradeoff,Advanced Regression Techniques (Ridge, Lasso, Elastic Net),Support Vector Machines (SVM) – Kernel methods, Ensemble Learning – Bagging, Boosting, Stacking, Gradient Boosting Algorithms (XGBoost, LightGBM, CatBoost), Model Evaluation Metrics (ROC, AUC, Precision-Recall, F1 Score), Cross Validation and Hyperparameter Tuning., Automated Machine Learning (AutoML)	12
II	**Advanced Unsupervised and Representation Learning** Clustering Techniques – K-Means, Hierarchical, DBSCAN, Gaussian Mixture Models (GMM), Dimensionality Reduction – PCA, LDA, t-SNE, Manifold Learning, Feature Engineering and Feature Selection, Anomaly Detection Techniques, Representation Learning.	12
III	**Deep Learning Architectures** Introduction to Deep Learning, Artificial Neural Networks (ANN), Backpropagation Algorithm, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), LSTM, GRU, Optimization	12

	Techniques (Adam, RMSProp), Regularization (Dropout, Batch Normalization),Transfer Learning and Fine-Tuning	
IV	**Advanced Topics in Machine Learning** Reinforcement Learning – Q-Learning, Policy Gradient,Markov Decision Process (MDP),Multi-Armed Bandit Problem,Graph Neural Networks (GNN),Federated Learning, Generative Models – GANs, Variational Autoencoders (VAE),Large Language Models (LLMs)	**12**
V	**Deployment, Scalability and Research Trends** MLOps – Model Deployment and Monitoring, Model Compression and Quantization, Edge AI,Distributed Machine Learning,Cloud-based ML Platforms,ML Security and Adversarial Attacks, TinyML	**12**
	Total	**60**

Text Books:

1. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 1st Edition, 2006. ISBN: 978-0387310732
2. Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, 1st Edition, 2016. ISBN: 978-0262035613
3. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow Aurélien Géron, O'Reilly Media, 2nd Edition, 2019. ISBN: 978-1492032649

Reference Books:

1. Machine Learning: A Probabilistic Perspective Kevin P. Murphy, MIT Press, 1st Edition, 2012. ISBN: 978-0262018029
2. Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, MIT Press, 2nd Edition, 2018. ISBN: 978-0262039246
3. Pattern Classification Richard O. Duda, Peter E. Hart, David G. Stork, Wiley-Interscience, 2nd Edition, 2000. ISBN: 978-0471056690

e-Books:

1. https://www.deeplearningbook.org
2. http://incompleteideas.net/book/the-book-2nd.html
3. https://d2l.ai/

MOOC / NPTEL/YouTube Links:

1. Machine Learning – NPTEL IIT Madras / Prof. S. Anandkumar
   https://onlinecourses.nptel.ac.in/noc22_cs70/preview
2. Reinforcement Learning – NPTEL IIT Madras
   https://onlinecourses.nptel.ac.in/noc20_cs91/preview
3. Machine Learning with Python – NPTEL (IIT Roorkee / IIT Guwahati)
   https://onlinecourses.nptel.ac.in/noc22_cs64/preview

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Computational Laboratory– I			Code	MCE26PCC-503		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
2	-	4	-	25	25	-	50

Pre-requisites: Prior knowledge of Python programming, Object-Oriented Programming concepts, Linear Algebra, and basic Machine Learning is required.

Course Objectives: This course aims at enabling students:

1. To understand advanced algorithmic techniques such as dynamic programming, backtracking, greedy methods, and graph traversal.
2. To Learn supervised, unsupervised, deep learning, and reinforcement learning models.
3. To explore the performance analysis of algorithms and machine learning models using evaluation metrics and execution time comparison.
4. To Learn optimization techniques, hyper parameter tuning, and model improvement strategies.
5. To understand model deployment, scalability, and modern trends like MLOps and TinyML.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Implement advanced algorithms such as Knapsack, LCS, N-Queen, BFS, DFS, and matrix multiplication.

CO2: Develop and compare regression, classification, clustering, and neural network models.

CO3: Analyse algorithm efficiency and evaluate machine-learning models using appropriate metrics.

CO4: Design reinforcement learning models and optimize deep learning architectures.

CO5: Deploy an end-to-end machine learning system with basic MLOps practices.

Guidelines for Students:

1. Solve any 4 assignments from each group A and Group B.
2. Maintain a proper lab record including aim, methodology, code, results, and performance analysis.
3. Perform comparative study of models using appropriate evaluation metrics.
4. Mini-project should demonstrate innovation, experimental validation, and research contribution.
5. Load the dataset from Kaggle or load it from scikit learn library.

Course Contents

Suggested List of Experiments/Assignments

Sl. No.	Group A: Advanced Algorithms
1	**Resource Allocation using Knapsack** A cloud service provider has limited server capacity. Each application requires a specific amount of resources and provides certain revenue. Implement the 0/1 Knapsack

	algorithm to determine the optimal allocation of applications that maximizes revenue without exceeding capacity. Compare the result with the Fractional Knapsack approach.
2	**DNA Sequence Matching using Dynamic Programming** Develop a program to compute the similarity between two DNA sequences. Implement the Longest Common Subsequence (LCS) algorithm using dynamic programming. Display the length of the longest common subsequence and the matched sequence.
3	**N-Queen Problem using Backtracking** Write a program to solve the N-Queen problem using the backtracking technique. The program should place N queens on an N×N chessboard such that no two queens attack each other. Display all possible valid solutions.
4	**Sequential vs Parallel Matrix Multiplication** Implement matrix multiplication in two ways: sequential execution and multithreaded execution. Compare the execution time for both implementations and analyze the speedup achieved using parallel processing.
5	**Network Connectivity Analysis** Given a graph representing a computer network, develop a program to check connectivity using Breadth First Search (BFS) and Depth First Search (DFS). Identify connected components and display traversal order.
	Group B: Advanced Machine Learning
1	**Regression and Classification Model Comparison.** Load a simple dataset e.g., housing price (Boston).apply Linear Regression, Ridge or Lasso Regression. Train a Support Vector Machine (SVM) classifier on a classification dataset.(Eg. Titanic)and Compare models using:Accuracy,F1 Score,ROC-AUC
2	**Customer Segmentation using Unsupervised Learning.** Use a customer dataset. Apply K-Means, Hierarchical Clustering, And DBSCAN algorithms and Compare cluster results using silhouette score. Write comparison analysis.
3	**Build and Optimize a Neural Network** Build an Artificial Neural Network (ANN) and Implement: Dropout, Batch Normalization with comparison like Adam,RMSProp of optimizers and apply transfer learning using a pre-trained CNN model.Evaluate performance using Precision, Recall and F1-score
4	**Implement Q-Learning for Game Simulation** Create a simple grid-world environment and Implement Q-Learning algorithm.Plot a graph for rewards and episodes.
5	**End-to-End ML System with MLOps Concepts** Build a classification model (any dataset).Perform hyperparameter tuning using GridSearch or AutoML.Deploy model using:Flask / FastAPI and write report on Model compression and TinyML concepts.
Reference Books: 1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, 2016.	

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Data Engineering			Code:		MCE26PEC-504A	
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of DBMS, and Software Engineering is required.

Course Objectives: This course aims at enabling Students

1. To introduce the foundational concepts and tools used in data engineering.
2. To understand the architecture of modern data pipelines and platforms.
3. To explore data ingestion, transformation, storage, and orchestration techniques.
4. To gain insight into scalable data processing using batch and stream frameworks.
5. To nable students to design reliable, secure, and ethical data pipelines and use Data Visualization tools.

Course Outcomes: After completion of the course, the students will be able to:

CO1. Preprocessing techniques for various datasets,
CO2. Design data pipelines for data ingestion, storage, and processing.
CO3. Apply data transformation and ETL principles using conceptual tools.
CO4. Sign and verify messages using different signature generation and verification algorithms.
CO5. Able to implement Visualization techniques and use DE knowledge in various applications

Course Contents

Unit	Description	Duration [Hrs]
I	**Introduction to Data Engineering** Introduction to data engineering, Types of data: text, video, audio, Healthcare data, Data Collection: Various sources of data. role in the AI/ML lifecycle, Categories: Structured, Semi-structured, Unstructured, data lifecycle, ingestion, storage, processing, analysis, scope and responsibilities in data engineering, data types handled in real-world systems, overview of data engineering roles and case studies of real-world examples,	9
II	**Data Storage** Data sources: databases, APIs, IoT, social media, Logs, batch and real-time ingestion, file systems, HDFS, object storage, S3, GCS, Azure clouds, Database, Schema, ER diagram, SQL, functions, stored procedures, indexing B+tree, MongoDB, Client-Server Architecture, relational, PostgreSQL, NoSQL, MongoDB, Cassandra, Data lakes and data warehouses, Concepts of data lakehouse, Delta Lake, Apache Iceberg, data ingestion and architectures for various use cases, storage solutions based on data structure and scale.	9

III	**Data Preprocessing & Optimization** Sanity check, Cleaning data, missing data, noisy data elimination, feature selection and dimensionality reduction, normalization, data modeling: Star and Snowflake schema, conceptual tools, ETL and ELT pipelines, concepts and use cases, data validation and quality checks Apache NiFi, Airbyte, dbt, Informatica, data transformation, data handling schema evolution, concept of underpinnings of ETL and transformation pipelines, evaluate data quality, consistency, and schema design.	**9**
IV	**Data Analysis and Distributed Data Processing** Regression, principal component analysis (PCA), canonical correlation analysis, analysis of variance. Distributed Data Processing: Need for distributed processing, batch processing, Hadoop, Apache Spark– architecture and concepts, stream processing, Apache Kafka, Flink, Spark Structured Streaming, data partitioning and shuffling, fault tolerance and parallelism, AI/ML pipelines– training large models, real-time ML systems workflows, Data orchestration, metadata management and lineage, data security, encryption, access control, anonymization,	**9**
V	**Data Visualization and Data Engineering Applications** Table, graph, histogram, pie-chart, area-plot, box-plot, scatter-plot, bubble-plot, waffle charts, word clouds, Applications: Recommendation systems, Time series analysis, E-commerce and agricultural AI pipelines, banking system- Fraud detection, Predictive analysis, use of pipelines in AI/ML applications	**9**
	Total	**45**

Text Books:

1. Joe Reis and Matt Housley, Fundamentals of Data Engineering, 1st Edition O'Reilly, July 2022, ISBN: 978-1-098-10830-4 https://freecomputerbooks.com/books/Fundamentals-of-Data-Engineering
2. Martin Kleppmann,"Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scal able, and Maintainable Systems", O'Reilly Media, Inc., 2017
3. Data Engineering: A Novel Approach to Data Design. Brian Shive. Technics Publications. ISBN-13: 978-1935504603

Reference Books:

1. Yupo Chan, John R. Talburt, Tery M. Tally, "Data Engineering: Mining, Information and Intelli gence", Springer, ISBN: 978-1-4419-0176-1
2. Paul Crickard, "Data Engineering with Python", Packt, ISBN: 978-1-83921-418-9, 2020
3. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools. Cielen, Meysman,Ali. Dreamtech Press. ISBN-13: 978-9351199373
4. A. Forouzan, "Cryptography & Network Security", Tata Mc Graw Hill.

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Computer Vision			Code:		MCE26PEC-504B	
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Digital Image Processing, Deep Learning is required.

Course Objectives: This course aims at enabling students:

1. To develop strong foundations in geometric image formation and multi-view vision.
2. To analyze early and mid-level vision algorithms using mathematical models.
3. To design robust segmentation, tracking, and reconstruction systems.
4. To implement object recognition pipelines combining geometry and learning.
5. To critically evaluate vision algorithms for research and industrial applications.

Course Outcome: After completion of the course, the students will be able to:

CO1: Model camera geometry and perform calibration and projection analysis.
CO2: Implement feature detection, matching, and segmentation algorithms.
CO3: Develop stereo and multi-view reconstruction systems.
CO4: Design object detection and recognition pipelines using modern classification strategies.
CO5: Analyze research problems in 3D vision, tracking, and high-level recognition.

Course Contents

Unit	Description	Duration [Hrs]
I	**Image Formation and Early Vision** Introduction to computer vision applications, image formation process, pinhole camera model, intrinsic and extrinsic parameters, basic camera calibration, light reflection and shading, color image basics, image filtering, Gaussian smoothing, image gradients, edge and corner detection.	9
II	**Features and Matching** Local feature detection, SIFT concept, HOG features, feature matching between images, template matching, image pyramids, basic image alignment, panorama stitching concept.	9
III	**Stereo and 3D Vision** Epipolar geometry concept, fundamental matrix idea, stereo vision basics, disparity and depth estimation, depth map generation, structure from motion concept, simple 3D reconstruction pipeline.	9
IV	**Segmentation and Tracking** Image segmentation basics, k-means clustering, watershed method, Hough transform, RANSAC concept, background subtraction, optical flow basics, object tracking using Kalman filter.	9

V	**Learning and Recognition** Image classification basics, training and testing datasets, support vector machine concept, bag of visual words model, sliding window object detection, face detection example, evaluation metrics such as accuracy and ROC curve.	**9**
	Total	**45**

Text Books:

1. David A. Forsyth and Jean Ponce, "Computer Vision: A Modern Approach", 2nd Edition, Pearson, 2012. ISBN: 9780136085928.
2. Richard Szeliski, "Computer Vision: Algorithms and Applications", Springer, 2010. ISBN: 9781848829343.

Reference Books:

1. Richard Hartley and Andrew Zisserman, "Multiple View Geometry in Computer Vision", 2nd Edition, Cambridge University Press, 2004. ISBN: 9780521540513.
2. Simon J. D. Prince, "Computer Vision: Models, Learning and Inference", Cambridge University Press, 2012. ISBN: 9781107011793.
3. Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", 4th Edition, Pearson, 2018. ISBN: 9780134449385

e-Books:

1. Richard Szeliski, "Computer Vision: Algorithms and Applications", freely available draft version at: https://szeliski.org/Book/

MOOC / NPTEL/YouTube Links:

1. Computer Vision – IIT Madras (NPTEL) https://onlinecourses.nptel.ac.in
2. Computer Vision – IIT Kharagpur (NPTEL) https://nptel.ac.in/courses
3. Introduction to Computer Vision – University at Buffalo (Coursera) https://www.coursera.org
4. Stanford CS231n – Convolutional Neural Networks for Visual Recognition https://cs231n.stanford.edu

Program	F.Y. M. Tech (Computer Engineering)			Semester: I			
Course	Advanced Computer Networks			Code:	MCE26PEC-504C		
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Basic Computer Networks and computer architecture is required.

Course Objectives: This course aims at enabling Students

1. To understand the principles of high-performance switching and routing including IP address lookup algorithms and hardware implementations.
2. To explore packet classification techniques, Quality of Service mechanisms, and traffic management including DiffServ, traffic policing, and traffic shaping.
3. To study Software Defined Networking (SDN) architecture, northbound and southbound interfaces, and hands-on experimentation using Mini net.
4. To understand Network Function Virtualization (NFV) architecture and programmable networks using P4 language and Smart NICs.
5. To analyze Data Center Networking topologies, Content Distribution architectures, Information-Centric Networking, and security in Named Data Networking.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Analyze and implement IP address lookup algorithms and evaluate performance trade-offs in high-performance switching and routing architectures.

CO2: Apply packet classification methods and Quality of Service mechanisms including Differentiated Services, traffic policing, and traffic shaping in network design.

CO3: Design and configure Software Defined Networking (SDN) environments using Mini net, Open Flow, and NFV architectures for real-world network scenarios.

CO4: Develop and test programmable data-plane applications using the P4 language on BMV2 software switches and Smart NIC platforms with Mini net.

CO5: Evaluate Data Center Network topologies, Content Distribution systems, Information-Centric Networking architectures, and security mechanisms in Named Data Networking.

Course Contents

Unit	Description	Duration [Hrs]
I	**High Performance Switching and Routing** Introduction to high-performance switching and routing; Performance considerations in routers and switches: throughput, latency, and scalability; IP address lookup: Exact match, Longest Prefix Match (LPM); Algorithms for IP address lookup: Trie-based (unibit, multibit), TCAM, hash-based approaches; Optimization techniques for lookup speed and memory efficiency; Hardware implementation of address	9

	lookup: pipelined architectures, parallel processing, FPGA and ASIC-based designs.	
II	**Packet Classification, QoS, and Network Softwarization** Need for packet classification: multi-field classification, policy enforcement, and firewall applications; Methods for packet classification: linear search, decision trees, tuple space search, hardware-based classification (TCAM); Differentiated Services (DiffServ): per-hop behaviors, PHB classes, DS codepoints; Quality of Service (QoS): integrated vs. differentiated services, SLA, QoS metrics; Traffic Policing: token bucket, leaky bucket algorithms, rate limiting; Traffic Shaping: techniques for smoothing burst traffic; Introduction to Network Softwarization: evolution from hardware-centric to software-driven networking.	**9**
III	**High Performance Switching and Routing** Introduction to high-performance switching and routing; Performance considerations in routers and switches: throughput, latency, and scalability; IP address lookup: Exact match, Longest Prefix Match (LPM); Algorithms for IP address lookup: Trie-based (unibit, multibit), TCAM, hash-based approaches; Optimization techniques for lookup speed and memory efficiency; Hardware implementation of address lookup: pipelined architectures, parallel processing, FPGA and ASIC-based designs.	**9**
IV	**Packet Classification, QoS, and Network Softwarization** Need for packet classification: multi-field classification, policy enforcement, and firewall applications; Methods for packet classification: linear search, decision trees, tuple space search, hardware-based classification (TCAM); Differentiated Services (DiffServ): per-hop behaviours, PHB classes, DS code points; Quality of Service (QoS): integrated vs. differentiated services, SLA, QoS metrics; Traffic Policing: token bucket, leaky bucket algorithms, rate limiting; Traffic Shaping: techniques for smoothing burst traffic; Introduction to Network Softwarization: evolution from hardware-centric to software-driven networking.	**9**
V	**High Performance Switching and Routing** Introduction to high-performance switching and routing; Performance considerations in routers and switches: throughput, latency, and scalability; IP address lookup: Exact match, Longest Prefix Match (LPM); Algorithms for IP address lookup: Trie-based (unibit, multibit), TCAM, hash-based approaches; Optimization techniques for lookup speed and memory efficiency; Hardware implementation of address lookup: pipelined architectures, parallel processing, FPGA and ASIC-based designs.	**9**
	Total	**45**

Text Books:

1. H. Jonathan Chao and Bin Liu, High Performance Switches and Routers, John Wiley & Sons, Inc., 2007, ISBN-10: 0-470-05367-4.
2. Gabriel M. de Brito, Pedro B. Velloso, Igor M. Moraes, Information-Centric Networks: A New Paradigm for the Internet, Wiley-ISTE, 1st Ed., 2013, ISBN: 9781848214491.
3. Larry Peterson, Carmelo Cascone, Brian O'Connor, Thomas Vachuska, and Bruce Davie, Software-Defined Networks: A Systems Approach, Online Free Reference, available at https://sdn.systemsapproach.org/index.html.

Reference Books:

1. B. Wissingh, C. Wood, A. Afanasyev, L. Zhang, D. Oran and C. Tschudin, Information-Centric Networking (ICN): CCNx and NDN Terminology, RFC 8793, June 2020.
2. Gary Lee, Cloud Networking: Understanding Cloud-based Data Centre Networks, Morgan Kaufmann, 2014, ISBN-13: 9780128007280.
3. Relevant Internet Drafts and RFCs from IETF (www.ietf.org); P4 Language Specification and tutorials at https://p4.org.

MOOC / NPTEL / e-sources:

1. NPTEL – Computer Networks: https://nptel.ac.in/courses/106105183 | SDN Course: https://nptel.ac.in/courses/106105080
2. P4 Language Tutorials and Documentation: https://p4.org/learn | Open Networking Foundation (ONF) SDN Resources: https://opennetworking.org
3. Named Data Networking Project Resources: https://named-data.net | Mininet Tutorials: http://mininet.org/walkthrough

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Cloud Computing			Code:		MCE26PEC-505A	
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Operating Systems and Computer Networks is required.

Course Objectives: This course aims at enabling Students

1. To understand cloud computing infrastructure, service models (IaaS, PaaS, SaaS), and gain hands-on experience with public cloud platforms such as AWS and Microsoft Azure.
2. To explore virtualization technologies including full virtualization, para-virtualization, VM snapshots, and cloning using hypervisor tools.
3. To set up and manage distributed data storage and processing systems using Hadoop HDFS, MapReduce, and Apache Spark.
4. To build, deploy, and orchestrate containerized applications using Docker and Kubernetes, applying best practices in container lifecycle management.
5. To implement security measures and sustainable practices in cloud environments, ensuring data protection, access control, and cost-efficient resource utilization.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Illustrate existing cloud hosting platforms, deployment models, and computing paradigms used in industry.

CO2: Describe cloud architecture, service models (IaaS/PaaS/SaaS), and major cloud platforms (AWS, Azure).

CO3: Apply virtualization techniques using VMware, AWS EC2, and Azure Virtual Machines.

CO4: Interpret sustainable data center design and apply Hadoop/MapReduce for large-scale data processing.

CO5: Demonstrate containerization-using Docker and evaluate cloud security mechanisms.

Course Contents

Unit	Description	Duration [Hrs]
I	**Introduction to Cloud Computing** Computing Evolution: Grid, Cluster, Utility, Autonomic; Cloud Characteristics and Definitions; Deployment Models: Public, Private, Hybrid, Community; Service Models: IaaS, PaaS, SaaS; Business Benefits; Green IT; Carbon Footprint in Cloud.	9
II	**Cloud Architecture and Service Management** Cloud Architecture and Reference Models; Elasticity and Scalability; Resource Management and Scheduling; QoS and SLAs; Cost Management; AWS: EC2, S3, SQS, ELB; Google App Engine; Microsoft Azure; Aneka; Cloud Platform Comparison.	9

III	**Virtualization Technologies** Virtualization Concepts; Server Consolidation; Types: Server, Storage, Network, Desktop; Full vs Para-virtualization; Hypervisors (Type-1 and Type-2); VMware Cloud; Hyper-V; Cloning, Snapshots, Templates; VM Migration; Green Data Centers: Design, Cooling, Energy Management. [Advanced Topics]: Unikernels (MirageOS, Unikraft) – lightweight VMs for security; Confidential Computing: Intel TDX, AMD SEV-SNP; Live VM Migration: pre-copy, post-copy, hybrid; Memory Over commitment and Balloon Drivers; NUMA-aware Scheduling; eBPF for cloud observability; Web Assembly (WASM) as next-gen container runtime; Liquid Cooling, Immersion Cooling, and AI-driven Power Usage Effectiveness (PUE) optimization in hyper scale data centers.	**9**
IV	**Cloud Data Storage and Big Data Processing** Hadoop Ecosystem; HDFS Architecture and Data Replication; MapReduce: Task and Data Partitioning; Hive and HBase; Apache Spark: RDD and Data Frame; Spark SQL; NoSQL Databases: MongoDB, Cassandra; Graph Databases: Neo4J; Data Synchronization.	**9**
V	**Cloud Security, Containers and Emerging Trends** Cloud Security Architecture; Threats and Vulnerabilities; IAM; Data Encryption; Docker: Dockerfile, Images, Containers, Docker Hub; Docker Compose; Kubernetes Overview; Microservices; Serverless (FaaS); Edge and Fog Computing; Cloud Migration Strategies; Future Trends. [Advanced Topics]: Cloud Security Posture Management (CSPM) and Cloud Workload Protection (CWPP); Zero Trust Architecture (ZTA) – NIST SP 800-207; FinOps: cloud cost governance and anomaly detection; Service Mesh: Istio and Linkerd for mTLS and observability; eBPF networking with Cilium for Kubernetes; Multi-Cloud and Hybrid Cloud: Anthos, Azure Arc, AWS Outposts; AI-driven Cloud Optimisation: predictive autoscaling; Sustainable Cloud: Carbon-Aware Computing, CNCF Sustainability frameworks.	**9**
	Total	**45**

Text Books:

1. Raj Kumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing: Principles and Paradigms, Wiley, 1st Ed. (2013).
2. Michael Miller, Cloud Computing: Web-Based Applications, Que Publishing, 1st Ed. (2008).
3. Tom E. White, Hadoop: The Definitive Guide, O'Reilly Media, 4th Ed. (2015).

Reference Books:

1. Anthony Velte, Toby Velte, Robert Elsenpeter, Cloud Computing: A Practical Approach, Tata McGraw-Hill, 3rd Ed. (2017).
2. Judith Hurwitz et al., Cloud Computing for Dummies, 1st Ed. (2009).

3. Bernd Oggl, Michael Kofler, Docker: Practical Guide for Developers and DevOps Teams, Rheinwerk Computing (2023).

MOOC / NPTEL / e-sources:

1. NPTEL – Cloud Computing: https://nptel.ac.in/courses/106105167
2. AWS Training – Cloud Practitioner Essentials: https://aws.amazon.com/training
3. Google Cloud Skills Boost: https://cloudskillsboost.google

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Cryptography and Cryptanalysis			Code:		MCE26PEC-505B	
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Discrete Mathematics and number theory is required.

Course Objectives: This course aims at enabling Students

1. To provide an introduction to the fundamental principles of cryptography and its applications on the network security domain.
2. To get familiar with cryptographic techniques for secure (confidential) communication of two parties over an insecure (public) channel;
3. To understand the role of cryptanalysis in the field of cryptography.
4. To communicate professionally about Cryptography and cryptaylysis.
5. To get familiar with cryptographic techniques for secure (confidential) communication of two parties over an insecure (public) channel.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Identify computer and network security threats and develop a security model to prevent, detect and recover from the attacks.
CO2: Design the security solutions for preventing the different security attacks
CO3: Apply different Cryptographic Techniques to encrypt and decrypt messages
CO4: Apply signature generation and verification algorithms to sign and verify message origin
CO5: Discuss different cryptanalysis techniques which can be applied in real time scenarios

Course Contents

Unit	Description	Duration [Hrs]
I	**Introduction to Cryptography and Cryptanalysis** Classical Encryption Techniques – Substitution Techniques, Transposition Techniques; Symmetric Cipher Model: Feistel cipher structure, DES,Tripple DES, Block Cipher Design Principles; AES	9
II	**Number Theory** Divisibility and the division Algorithm, The Euclidean Algorithm, Modular Arithmetic, Fermat's and Euler's Theorems, The Chinese reminder Theorem.	9
III	**Public Key Cryptography** RSA – Algorithm & Computational Aspects, Diffie- Hellman Key Exchange; Elgamal Cryptographic System; Elliptic Curve Cryptography	9
IV	**Key Management, Hash Functions and Authentication** Symmetric key distribution using symmetric & asymmetric encryption, distribution of public keys, X.509 certificates, PKI, Applications, SHA,	9

	MD5; Message Authentication Codes: requirements, function, security, HMAC; Digital signatures - introduction	
V	**Cryptanalysis** Cryptanalysis on Substitution Cipher (Frequency Analysis), Cryptanalysis on Stream Cipher, Modern Stream Ciphers, Time-Memory Trade-off Attack, Linear Cryptanalysis, Differential Cryptanalysis	**9**
	Total	**45**

Text Books:

1. William Stallings, "Crpyptography and Network security -Principles and Practices", Pearson publication sixth Edition.
2. Atul Kahate, "Crpyptography and Network security ", McGrawHill publication

Reference Books:

1. William Stallings, Lawrie Brown "Computer security -Principles and Practices", Pearson publication.
2. John F. Dooley, History of Cryptography & Cryptanalysis-Codes, Ciphers & Algorithms, Springer
3. W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education.
4. A. Forouzan, "Cryptography & Network Security", Tata Mc Graw Hill.

MOOC / NPTEL / e-sources:

1. NPTEL – Foundations of Cryptography (IIIT Bangalore): https://onlinecourses.nptel.ac.in/noc26_cs18/preview
2. NPTEL – Cryptography and Network Security (IIT Kharagpur): https://onlinecourses.nptel.ac.in/noc26_cs57/preview
3. Coursera – Cryptography I (Stanford University): https://www.coursera.org/learn/crypto
4. SWAYAM / IGNOU – Fundamental of Cryptography: https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ma04

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Deep Learning			Code:		MCE26PEC-505C	
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Probability & Statistics, and basic Machine Learning concepts is required.

Course Objectives: This course aims at enabling students:

1. To understand the mathematical foundations of Deep Learning.
2. To learn various neural network architectures and training techniques.
3. To explore advanced deep learning models such as CNNs, RNNs, and Transformers.
4. To develop and apply generative models (Autoencoders, VAE, GANs) and reinforcement learning techniques.
5. To implement deep learning algorithms using modern frameworks and follow ethical AI practices for real-world applications.

Course Outcome: After completion of the course, the students will be able to:

CO1: Explain fundamental concepts of artificial neural networks, optimization techniques, and regularization strategies.

CO2: Design and implement deep feedforward and convolutional neural networks for computer vision tasks.

CO3: Analyze sequence models such as RNN, LSTM, GRU, attention mechanisms, and Transformer architectures for NLP and speech processing.

CO4: Develop and apply generative models and reinforcement learning techniques for research and practical applications.

CO5: Apply deep learning frameworks (TensorFlow, PyTorch) to build, train, optimize, and deploy models while considering ethical and responsible AI practices.

Course Contents

Unit	Description	Duration [Hrs]
I	**Foundations of Deep Learning** Introduction to AI, ML and Deep Learning. Biological vs Artificial Neuron. Perceptron and Multilayer Perceptron (MLP). Activation Functions (Sigmoid, Tanh, ReLU, Softmax). Loss Functions. Gradient Descent and Variants (SGD, Momentum, RMSProp, Adam). Backpropagation Algorithm. Regularization techniques (L1, L2, Dropout).	9
II	**Convolutional Neural Networks (CNNs)** Convolution Operation, Padding, Stride, Pooling. CNN Architectures: LeNet, AlexNet, VGG, ResNet. Transfer Learning and Fine-tuning. Batch Normalization. Object Detection and Image Segmentation basics. Applications in Computer Vision.	9

III	**Sequence Models and Attention Mechanisms** Recurrent Neural Networks (RNN). Vanishing/Exploding Gradient Problem. LSTM and GRU architectures. Bidirectional RNN. Encoder-Decoder Models. Attention Mechanism. Transformer Architecture basics. Applications in NLP and Speech Processing.	**9**
IV	**Generative Models and Advanced Architectures** Autoencoders, Sparse Autoencoders. Variational Autoencoders (VAE). Generative Adversarial Networks (GAN), Conditional GAN. Deep Reinforcement Learning basics. Model evaluation metrics. Hyperparameter tuning strategies.	**9**
V	**Deep Learning Frameworks and Applications** TensorFlow and PyTorch fundamentals. Building, training and evaluating models. Model optimization and deployment basics. Case Studies: Image Classification, Text Classification, Sentiment Analysis. Ethical Issues in AI and Responsible AI practices.	**9**
	Total	**45**

Text Books:

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning", MIT Press, 2016. ISBN: 9780262035613.
2. Simon Haykin, "Neural Networks and Learning Machines", 3rd Edition, Pearson, 2009. ISBN: 9780131471399.

Reference Books:

1. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006. ISBN: 9780387310732.
2. François Chollet, "Deep Learning with Python", Manning, 2nd Edition, 2021.
3. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow", O'Reilly, 3rd Edition, 2022.

e-Books:

1. https://www.deeplearningbook.org

MOOC / NPTEL/YouTube Links:

1. Deep Learning – IIT Ropar / IIT Madras
   https://onlinecourses.nptel.ac.in/noc26_cs66/preview
   https://onlinecourses-archive.nptel.ac.in/noc18_cs41/preview
2. Deep Learning – IIT Kharagpur (Machine Learning + Deep Learning focus)
   https://onlinecourses.nptel.ac.in/noc26_ee53/preview
3. Deep Learning – Swayam Course
   https://nptel.ac.in/courses/106106184

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Project Based Learning			Code	MCE26PEC-506		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
2	-	4	-	25	25	-	50

Pre-requisites: Prior knowledge of Programme Elective Course is required.

Course Objectives: This course aims at enabling students:

1. To formulate research problems based on selected Programme Elective (PEC-I and PEC-II) domains.
2. To design and implement experimental or prototype-based solutions addressing real-world or research challenges.
3. To apply scientific methodology for performance evaluation, comparative analysis, and validation of results.
4. To develop research aptitude through literature survey, innovation, technical documentation, and proper citation practices.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Identify and define research-oriented problem statements from chosen elective domains.

CO2: Design and develop experimental frameworks or system prototypes with measurable performance metrics.

CO3: Analyse and validate results using benchmarking, statistical methods, or comparative evaluation techniques.

CO4: Present research findings in structured technical documentation aligned with academic and publication standards.

Guidelines for Students:

1. The laboratory shall be conducted strictly in **Project-Based Learning (PBL)** mode and will consist of **two research-oriented mini projects**:
    - Mini Project 1: Based on **PEC–I**
    - Mini Project 2: Based on **PEC–II**
2. Students must select problem statements derived from contemporary research challenges or real-world industry problems relevant to their chosen electives.
3. Each mini project must include:
    - Literature survey of minimum **15 research papers**
    - Problem definition and objective formulation
    - System design and methodology
    - Implementation and experimentation
    - Performance evaluation using appropriate metrics
    - Comparative analysis with baseline methods
4. Projects must demonstrate **innovation, optimization, enhancement, or measurable improvement** over existing approaches.
5. Quantitative evaluation (accuracy, latency, throughput, security strength, scalability, etc., as applicable) is mandatory.

6. Students must maintain proper documentation including:

    - Literature review summary
    - System architecture
    - Experimental setup
    - Results and analysis
    - Conclusion and future scope
    - Proper citation in IEEE/APA format
7. Each mini project should aim to produce a **conference-paper-level technical report** or research prototype.
8. Plagiarism in code, datasets, or documentation is strictly prohibited. All external references must be properly cited.
9. Demonstration and viva shall evaluate:
    - Conceptual clarity
    - Research contribution
    - Experimental validation
    - Understanding of domain-specific challenges

Phase I – Foundation and Problem Understanding
Introduction to the elective domain, overview of core concepts and terminology, identification of real-world applications, understanding system architecture and workflow, study of fundamental models and techniques, analysis of existing tools and technologies, discussion of domain-specific challenges and limitations
Phase II – Methods and Techniques
Study of major algorithms and methodologies used in the domain, comparative analysis of classical and modern approaches, understanding design principles, modelling techniques, performance parameters, implementation frameworks, simulation or experimental setup considerations.
Phase III – System Design and Implementation
Problem statement formulation, requirement analysis, system architecture design, selection of appropriate tools and platforms, dataset or input preparation, development of prototype or experimental framework, integration of modules, testing strategies
Phase IV – Performance Evaluation and Optimization
Definition of evaluation metrics, benchmarking with baseline methods, experimental validation, result analysis, statistical comparison, optimization techniques, scalability considerations, robustness and reliability testing.
Phase V – Advanced Topics and Research Directions
Recent advancements in the domain, emerging technologies and trends, research challenges, interdisciplinary applications, ethical and societal considerations, innovation opportunities, preparation of technical documentation, presentation and publication standards.

Program	F.Y. M.Tech (Computer Engineering)			Semester: I			
Course	Research Methodology			Code:		MCE26ELC -507	
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
4	4	-	-	25	25	50	100

Pre-requisites: Prior knowledge of engineering concepts, elementary statistics, and fundamental technical writing skills is required.

Course Objectives: This course aims at enabling students:

1. To understand the scientific method of conducting research and develop systematic research skills.
2. To learn various research methodologies, data collection, and analysis techniques.
3. To develop competence in technical report writing and research publication.
4. To understand Intellectual Property Rights (IPR), publication ethics, and startup potential of innovative ideas.

Course Outcome: After completion of the course, the students will be able to:

CO1: Apply fundamental research concepts and methodologies to identify and define research problems.

CO2: Design appropriate research methods, collect and analyze data systematically.

CO3: Prepare research reports and technical papers following ethical standards and publication guidelines.

CO4: Demonstrate understanding of research dissemination through journals, conferences, IPR filing, and startup initiatives.

CO5: Evaluate research problems critically and propose innovative solutions with consideration of ethical, societal, and entrepreneurial implications.

Course Contents

Unit	Description	Duration [Hrs]
I	**Introduction to Research Methodology** Meaning and definition of research, objectives and significance of research, characteristics of good research, types of research such as basic research, applied research, exploratory research, descriptive research and analytical research, scientific method and research process, identification and formulation of research problem, defining research objectives and scope, literature review techniques and sources including journals, conferences and digital libraries, identification of research gaps, hypothesis formulation, research variables and assumptions.	12
II	**Research Design and Data Collection Methods** Concept and need of research design, types of research design including experimental design, survey design and case study method, qualitative and quantitative research approaches, sampling theory and sampling techniques including probability sampling and non-probability	12

	sampling, sample size determination, methods of data collection including primary and secondary data, questionnaire design and scaling techniques, interview method and observation method, reliability and validity of research instruments, pilot study, ethical issues in research, plagiarism and research integrity.	
III	**Data Analysis and Interpretation** Data preparation and data cleaning, coding and classification of data, tabulation and graphical representation, descriptive statistics including mean, median, mode, range and standard deviation, inferential statistics including correlation analysis, regression analysis and hypothesis testing, chi-square test, t-test and ANOVA basics, use of statistical software tools such as MS Excel and SPSS basics, interpretation of statistical results, drawing conclusions and recommendations based on analysis.	**12**
IV	**Research Reporting and Publication** Structure and components of research report including abstract, introduction, methodology, results and conclusion, thesis and dissertation writing guidelines, technical paper writing format, preparation of research articles for journals and conferences, referencing styles such as IEEE and APA format, citation management tools, publication ethics and avoidance of plagiarism, peer review process, impact factor and indexing databases, presentation skills for seminars and conferences, preparation of posters and research presentations.	**12**
V	**Intellectual Property Rights (IPR) and Start-up Process** Introduction to Intellectual Property Rights, need and importance of IPR, types of intellectual property including patents, copyrights, trademarks and industrial designs, patentability criteria and patent search process, procedure for patent filing in India, drafting of patent specification, technology transfer and commercialization of research outcomes, innovation management, startup ecosystem in India, incubation centres and funding agencies, converting research ideas into start-ups, basics of entrepreneurship development and business model preparation.	**12**
	Total	**60**

Text Books:

1. C.R. Kothari and Gaurav Garg, "Research Methodology: Methods and Techniques", 4th Edition, New Age International Publishers, 2019, ISBN: 9789386649225.
2. John W. Creswell and J. David Creswell, "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches", 5th Edition, SAGE Publications, 2018, ISBN: 9781506386706.
3. Uma Sekaran and Roger Bougie, "Research Methods for Business: A Skill-Building Approach", 7th Edition, Wiley India, 2016, ISBN: 9788126559015.

Reference Books:

1. Ranjit Kumar, “Research Methodology: A Step-by-Step Guide for Beginners”, 5th Edition, SAGE Publications, 2019, ISBN: 9781526449903.
2. Wayne C. Booth, Gregory G. Colomb, and Joseph M. Williams, “The Craft of Research”, 4th Edition, University of Chicago Press, 2016, ISBN: 9780226239736.
3. Government of India, “Manual of Patent Practice and Procedure”, The Office of the Controller General of Patents, Designs and Trade Marks, 2019 Edition.

e-Books:

1. https://www.euacademic.org/BookUpload/9.pdf

MOOC / NPTEL / e-sources:

1. NPTEL course on Research Methodology by Prof. Soumitro Banerjee, IISER Kolkatahttps://onlinecourses.nptel.ac.in/noc22_ge08/preview
2. Coursera course on Research Methodologies by Athanasia
3. Lamprakihttps://www.coursera.org/learn/research-methodologies

Course Syllabus

Semester-II

Program	F.Y. M.Tech (Computer Engineering)			Semester : II			
Course	Advanced Databases			Code:	MCE26PCC-551		
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
4	4	4	-	25	25	50	100

Pre-requisites: Prior knowledge of data structures and Database Management Systems fundamentals is required.

Course Objectives: This course aims at enabling students:

1. To understand the basic concepts, architecture, and design approaches of distributed database systems.
2. To learn data fragmentation, allocation strategies, and transparency concepts in distributed environments.
3. To study data management, query processing, and security techniques in distributed databases.
4. To understand concurrency control mechanisms and distributed transaction management protocols.
5. To learn parallel database architectures and techniques for parallel query processing and performance improvement.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Explain the fundamentals of distributed database systems, including architecture, design approaches, fragmentation, allocation, transparency, and the CAP theorem.

CO2: Apply data management techniques such as access control, view management, query processing, and encryption for secure and efficient distributed databases.

CO3: Analyse and optimize distributed queries using decomposition, rewriting, join strategies, and optimization techniques.

CO4: Evaluate concurrency control and distributed transaction protocols like Two-Phase and Three-Phase Commit, along with replication and failure handling.

CO5: Assess parallel database architectures, query processing, load balancing, and performance methods to enhance scalability and efficiency.

Course Contents

Unit	Description	Duration [Hrs]
I	**Fundamentals of Distributed Database Systems** Introduction to Distributed Database Systems, Advantages and challenges of distributed databases, Distributed database system architecture, Top-down approach to database design, Bottom-up approach to database design, Distributed database design issues, Data fragmentation (horizontal, vertical, derived), Data allocation strategies, Database integration, Schema matching, schema integration, schema mapping, Transparency in distributed systems	12

	(location, replication, fragmentation transparency), CAP theorem basics and its importance in distributed systems.	
II	**Data Management and Query Processing** Data and access control in distributed databases, View management, Data security in distributed environment, Query processing problems, Objectives of query processing, Complexity of relational algebra operations, Characterization of query processors, Layers of query processing, Encryption techniques for distributed data protection	**12**
III	**Distributed Query Optimization and Execution** Query decomposition, Query normalization and analysis, Elimination of redundancy and query rewriting, Localization of distributed data, Reduction techniques for primary horizontal, vertical, and derived fragmentation, Distributed query execution, Query optimization techniques, Join ordering strategies, Static and dynamic query optimization approaches, Semi-joins and hybrid join approaches	**12**
IV	**Concurrency Control and Distributed Transactions** Taxonomy of concurrency control mechanisms, Lock-based concurrency control, Timestamp-based concurrency control, Optimistic concurrency control, Deadlock management techniques, Heterogeneity issues in distributed systems, Advanced transaction models, Two-Phase Commit (2PC) protocol, Three-Phase Commit (3PC) protocol, Replication protocols, Replication and failure handling, Multiversion concurrency control (MVCC)	**12**
V	**Parallel Databases** Introduction to parallel databases, Parallel database system architectures (shared memory, shared disk, shared nothing), Parallel data placement, Full partitioning, Parallel query processing, Query parallelism (intra-query and inter-query), Parallel query optimization, Load balancing techniques, Performance evaluation and benchmarking of parallel databases	**12**
	Total	**60**

Text Books:

1. M T Ozsu, Patrick Valduriez, Principles of Distributed Database Systems, Prentice Hall, 1999.
2. S. Ceri and G. Pelaggati, Distributed Database System Principles and Systems, MGH, 1985.
3. Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", McGraw Hill Publishers, 7th Edition, 2020 ISBN 978-0-07-802215-9.
4. Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle", BPB Publications, 2014 ISBN: 9788176569644.
5. Pramod J. Sadalage and Martin Fowler, "NoSQL Distilled", Addison Wesley, ISBN 10: 0321826620, 2013, ISBN 13: 978-0321826626.
6. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, McGraw Hill Education, 3rd edition (2014), ISBN-10: 9339213114, ISBN-13: 978- 9339213114

Reference Books:

1. C. J. Date, "An Introduction to Database Systems", Addison-Wesley, 8th Edition, 2004, ISBN 0321189566. S. K. Singh, "Database Systems: Concepts, Design and Application", Pearson Education, 2009, ISBN 9788177585674.Kristina Chodorow, Michael Dierolf, "MongoDB: The Definitive Guide", O‟Reilly Publications, 3rd Edition, 2019 ISBN 9781491954461.
2. Kevin Roebuck, "Storing and Managing Big Data - NoSQL, HADOOP and More", Emereo Pty Limited, 2011, ISBN 1743045743, 9781743045749.
3. Korth, Silberschatz and Sudarshan, “Database System Concepts”, Tata McGraw Hill, 6th edition (2013), ISBN-10: 9332901384, ISBN-13: 978-9332901384
4. R. Elmasri, and S. Navathe, “Fundamentals of Database Systems”, Pearson, 7th edition (2017), ISBN-10: 9789332582705, ISBN-13: 978-9332582705

e-Books:

1. https://www.perlego.com/book/4500974/advanced-database-systems-pdf
2. https://www.perlego.com/book/4500859/advanced-data-management-for-sql-nosql-cloud-and-distributed-databases-pdf

MOOC / NPTEL/YouTube Links:

1. https://onlinecourses.nptel.ac.in/noc25_cs84/preview
2. https://onlinecourses.nptel.ac.in/noc25_cs40/preview
3. https://archive.nptel.ac.in/courses/106/105/106105175/

Program	F.Y. M.Tech (Computer Engineering)			Semester: II			
Course	Quantum Computing			Code:	MCE26PCC-552		
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
4	4	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Computer networks and fundamental security concepts is required.

Course Objectives: This course aims at enabling Students

1. To being able to analyze simple quantum algorithms and argue optimality.
2. To familiarity with 1-qubit / 2-qubit gate operators and ability to design simple quantum circuits.
3. To ability to read and understand recent results as well as research papers on quantum algorithms

Course Outcomes: After completion of the course, the students will be able to:

CO1: Explain how the physics of quantum computation is different from classical computational models.

CO2: Describe the theoretical performance improvements that quantum algorithms offer com- pared to classical algorithms.

CO3: Analyze the life cycle of hybrid applications and decompose their execution on a hybrid quantum-classical computational continuum.

CO4: Develop their own (hybrid) quantum algorithms and implement them using (real or simulated) quantum computers using quantum toolkits such as Qiskit.

Course Contents

Unit	Description	Duration [Hrs]
I	**Introduction and the leap from Classical to Quantum** Introduction: Cbits and Qbits, Reversible and Manipulating operations on Cbits and Qbits, Circuit diagrams, Measurement gates and state preparation, Constructing arbitrary 1- and 2-Qbit states, The Leap from Classical to Quantum: Classical Deterministic Systems, Probabilistic Systems, Quan- tum Systems, Assembling Systems	9
II	**Basic Quantum Theory & Architecture** Basic Quantum Theory: Quantum States, Observables, Measuring, Dynamics, Assembling Quantum Systems Architecture: Bits and Qubits, Classical Gates, Reversible Gates, Quantum Gates	9
III	**Algorithms & Programming Languages** Algorithms: Deutsch's Algorithm, The Deutsch–Jozsa Algorithm, Simon's Periodicity Algorithm, Grover's Search Algorithm, Shor's Factoring Algorithm,	9

	Programming Languages: Programming in a Quantum World, Quantum Assembly Programming, Toward Higher-Level Quantum Programming, Quantum Computation Before Quantum Computers	
IV	**Theoretical Computer Science and Cryptography** Theoretical Computer Science: Deterministic and Nondeterministic Computations, Probabilistic Computations, Quantum Computations Cryptography: Classical Cryptography, Quantum Key Exchange I: The BB84 Protocol, Quantum Key Exchange II: The B92 Protocol, Quantum Key Exchange III: The EPR Protocol, Quantum Teleportation	**9**
V	**Information Theory & Hardware** Information Theory: Classical Information and Shannon Entropy, Quantum Information and von Neumann Entropy, Classical and Quantum Data Compression, Error-Correcting Codes Hardware: Quantum Hardware: Goals and Challenges, implementing a Quantum Computer I: Ion Traps, implementing a Quantum Computer II: Linear Optics, Implementing a Quantum Computer III: NMR and Superconductors, Future of Quantum Ware	**9**
	Total	**45**

Text Books:

1. Eleanor Rieffel and Wolfgang Polak, "Quantum Computing: A Gentle Introduction," The MIT Press, 2011, ISBN 978-0-262-01506-6
2. Quantum Computation and Quantum Information by Nielsen and Chuang (NC), Classical and Quantum Computation by Kitaev, Shen, and Vyalyi (KSV)
3. Scott Aaronson, "Quantum Computing Since Democritus," Cambridge University Press, March 2013, ISBN: 9780521199568

Reference Books:

1. N. David Mermin, "Quantum Computer Science: An Introduction," Cambridge University Press, August 2007, ISBN: 9780521876582
2. Noson S. Yanofsky and Mirco A. Mannucci, "Quantum Computing for Computer Scientists," Cambridge University Press, August 2008, ISBN: 9780521879965
3. "IBM Quantum Computing- Qiskit - ibm.com," https://www.ibm.com/quantum/qiskit

Program	F.Y. M.Tech (Computer Engineering)			Semester: II			
Course	Natural Language Processing			Code:	MCE26PCC-553		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
4	4	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Machine Learning and Data Science is required.

Course Objectives: This course aims at enabling students:

1. To understand the fundamental concepts and techniques of Natural Language Processing.
2. To explore the levels of linguistic analysis: Morphology, Syntax, Semantics, and Pragmatics.
3. To learn probabilistic and statistical models for language processing.
4. To gain expertise in modern Deep Learning architectures (Transformers, BERT) for NLP.
5. To design and implement NLP applications like Machine Translation, Sentiment Analysis, and Chabot's.

Course Outcome: After completion of the course, the students will be able to

CO1: Apply pre-processing techniques to clean and prepare text data for modeling

CO2: Analyze the syntactic and semantic structure of sentences using formal grammars and parsers.

CO3: Implement probabilistic models (HMM, N-grams) for sequence labeling and prediction.

CO4: Develop and fine-tune State-of-the-Art (SOTA) Deep Learning models for complex NLP tasks.

CO5: Evaluate the performance of NLP systems using standard metrics like BLEU, ROUGE, and F1-score.

Course Contents

Unit	Description	Duration [Hrs]
I	**Introduction & Text Pre-processing** Foundations: History of NLP, Generic NLP Pipeline, Challenges (Ambiguity, Diversity). Morphology: Tokenization, stemming vs. Lemmatization, Stop-word removal. Regular Expressions: Pattern matching for text clean up and normalization. Edit Distance: Levenshtein distance and spell-checking algorithms.	9
II	**Language Modelling & Sequence Labeling** N-Grams: Unigrams, Bigrams, Trigrams; Language modeling and Smoothing (Laplace, Good-Turing). POS Tagging: Parts of Speech, Tag sets (Penn Treebank), Rule-based vs. Stochastic tagging. Hidden Markov Models (HMM): Viterbi Algorithm for sequence labeling. NER: Named Entity Recognition and its applications in Information Extraction.	9

III	**Syntax and Semantic Analysis** **Parsing:** Context-Free Grammars (CFG), CKY Algorithm, Earley Parsing. **Dependency Parsing:** Transition-based and Graph-based parsing. **Word Senses:** WordNet, Polysemy, Word Sense Disambiguation (WSD).**Vector** **Semantics:** TF-IDF, Word2Vec (Skip-gram & CBOW), GloVe, and FastText.	**9**
IV	**Deep Learning for NLP** Recurrent Neural Networks (RNN): Vanishing Gradient problem, LSTMs, and GRUs.Encoder-Decoder Architecture: Sequence-to-Sequence (Seq2Seq) models. Attention Mechanism: Scaled Dot-Product Attention, Multi-head Attention. The Transformer: Architecture of "Attention is All	**9**
V	**Advanced Topics & Applications** Large Language Models (LLMs): BERT (Bi-directional Encoder), GPT (Generative Pre-Trained Transformer), Roberta. Fine-tuning: Transfer Learning in NLP, PEFT (Parameter Efficient Fine-Tuning).Applications: Neural Machine Translation (NMT), Summarization, Sentiment Analysis, Question Answering systems, and Ethics in NLP (Bias & Fairness).	**9**
	Total	

Text Books:

1. Daniel Jurafsky and James H. Martin, "Speech and Language Processing", Pearson (3rd Edition Draft).
2. Christopher Manning and Hinrich Schütze, "Foundations of Statistical Natural Language Processing", MIT Press.
3. Yoav Goldberg, "Neural Network Methods for Natural Language Processing", Morgan & Claypool.

Reference Books:

1. Natural Language Processing with Transformers – Lewis Tunstall, Leandro von Werra, and Thomas Wolf. (Essential for Hugging Face implementation).
2. Neural Network Methods in Natural Language Processing – Yoav Goldberg. Great for deep learning theory.
3. Build a Large Language Model (from Scratch) – Sebastian Raschka (2025). Highly recommended for modern M.Tech projects

e-Books:

1. https://web.stanford.edu/~jurafsky/slp3/
2. https://nlp.stanford.edu/fsnlp/
3. https://github.com/rasbt/LLMs-from-scratch
4. https://github.com/nlp-with-transformers/notebooks

MOOC / NPTEL/YouTube Links:

1. https://nptel.ac.in/courses/106105158
2. https://onlinecourses.nptel.ac.in/noc26_cs88/preview
3. https://onlinecourses.nptel.ac.in/noc26_cs33/preview

Program	F.Y. M.Tech (Computer Engineering)			Semester: II			
Course	Computational Laboratory– II			Code	MCE26PCC-554		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
2	-	4	-	25	25	-	50

Pre-requisites: Prior knowledge of Advanced Databases, Quantum Computing, and Natural Language Processing is required.

Course Objectives: This course aims at enabling students:

1. To implement, design, and optimize Deep Learning and NLP models using modern frameworks for real-world applications.
2. To scientifically analyse, evaluate, and compare model architectures and system performance across AI and data-driven environments.
3. To develop research-oriented problem-solving skills, encouraging innovation, experimentation, and prototype or publication readiness.
4. To acquire foundational knowledge of quantum computing, including quantum algorithms, 1-qubit and 2-qubit gates, and simple quantum circuit design.
5. To understand and implement advanced and distributed NoSQL database systems (e.g., MongoDB, Cassandra) with focus on scalability, indexing, replication, sharing, consistency, and fault tolerance.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Design and implement intelligent systems using Deep Learning frameworks (TensorFlow/PyTorch) and distributed NoSQL databases (MongoDB/Apache Cassandra) with appropriate data modeling and optimization techniques.

CO2: Develop and optimize advanced architectures including CNN, RNN, LSTM, Transformer models, and design simple quantum circuits using 1-qubit and 2-qubit gate operators.

CO3: Analyze and evaluate system performance through experimental studies, performance metrics, comparative analysis, and assessment of distributed database consistency and scalability.

CO4: Apply quantum algorithms, advanced database concepts, and deep learning techniques to build integrated, real-world research prototypes following sound software engineering practices.

CO5: Prepare technical documentation, experimental reports, and research-oriented project summaries suitable for publication, conference submission, or prototype deployment.

Guidelines for Students:

1. Solve any 4 assignments from each Group A, B, C.
2. Maintain a proper lab record including aim, methodology, code, results, and performance analysis.
3. Perform comparative study of models using appropriate evaluation metrics.
4. Mini-project should demonstrate innovation, experimental validation, and research contribution.

5. All experiments must be implemented using Python and relevant frameworks such as TensorFlow/PyTorch wherever applicable.	
Course Contents	
Suggested List of Experiments/Assignments	
Sl. No.	**Group A: Advanced Databases**
1	Install and configure MongoDB environment and study its architecture including components such as database, collections, documents, storage engine, and server processes.
2	Design and implement a document-oriented data model in MongoDB using collections and embedded documents for efficient data representation.
3	Perform advanced Create, Read, Update, and Delete (CRUD) operations and analyse query optimization techniques in MongoDB.
4	Implement various indexing strategies in MongoDB and evaluate their impact on query performance using execution statistics.
5	Implement various indexing strategies in MongoDB and evaluate their impact on query performance using execution statistics.
6	Install Apache Cassandra and configure a multi-node cluster environment for distributed database operations.
7	Design and implement a column-family data model in Apache Cassandra using key spaces, tables, and partition keys
8	Study different consistency levels and replication strategies in Apache Cassandra and evaluate their effect on system performance.
	Group B: Quantum Computing
1	Write down the state vector (matrix representation) of two qubits, i.e. the tensor product, in the computational basis. Write down the basis vectors of the composite system
2	Write down the unitary matrix representations of the CNOT in the computational basis with qubit 1 being the control qubit. Write down the matrix in the same basis with qubit 2 being the control bit
3	How to transfer an unknown quantum state only using one entangled pair of qubits and classical information as a resource.
4	Create a single qubit initialized to ket 0 and ket 1 and compare classical NOT and quantum X gate using Qiskit.
5	Check the Matrix Representation and Verification of the CNOT Gate in Different Control Configurations
6	Analyze the Action of CNOT on Arbitrary States and Entanglement of Qubits
Group C: Natural Language Processing	
1	Analyze the Action of CNOT on Arbitrary States and Entanglement of Qubits

2	Compare Porter Stemmer vs. WordNet Lemmatizer on a diverse corpus. Analyze cases where stemming fails to preserve semantic meaning.
3	Implement the Levenshtein Distance (Edit Distance) algorithm using Dynamic Programming to build a "Did you mean?" spell-correction feature.
4	Build an N-Gram Language Model (Bigram/Trigram). Calculate the probability of a sentence and its Perplexity
5	Implement Laplace (Add-1) and Good-Turing Smoothing to handle the "Zero Probability" problem in your N-Gram model.
6	Develop a Hidden Markov Model (HMM) for Part-of-Speech (POS) tagging using the Viterbi Algorithm on the Penn Treebank dataset.

Reference Books:

1. Kristina Chodorow, MongoDB: The Definitive Guide, O'Reilly Media.Ivan Bayross,
2. Pramod J. Sadalage and Martin Fowler, "NoSQL Distilled", Addison Wesley, ISBN 10: 0321826620, 2013, ISBN 13: 978-0321826626.
3. Jeff Carpenter and Eben Hewitt, Cassandra: The Definitive Guide, O'Reilly Media.
4. Christopher Manning and Hinrich Schütze, "Foundations of Statistical Natural Language Processing", MIT Press.
5. Yoav Goldberg, "Neural Network Methods for Natural Language Processing", Morgan & Claypool.

Program	F.Y. M.Tech (Computer Engineering)			Semester: II			
Course	DevOps Engineering			Code:		MCE26PEC-555A	
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Cloud Computing and Computer Networks is required.

Course Objectives: This course aims at enabling Students

1. To understand the DevOps philosophy, CAMS culture, and end-to-end software delivery pipeline, with alignment to industry standards such as the IBM DevOps Professional Certificate, Google SRE, and Linux Foundation LFS261 curricula.
2. To implement version control using Git and build automated CI/CD pipelines with Jenkins and GitHub Actions for continuous integration, testing, and deployment.
3. To containerize applications using Docker and orchestrate them at scale with Kubernetes, applying best practices in container lifecycle and cluster management.
4. To provision and manage cloud infrastructure using Infrastructure as Code (IaC) tools such as Terraform and Ansible, enabling repeatable and scalable deployments.
5. To establish observability and security practices in DevOps pipelines using Prometheus, Grafana, and the ELK Stack for monitoring, along with DevSecOps principles for integrating security throughout the software delivery lifecycle.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Understand DevOps culture, CAMS principles, Agile-DevOps alignment, and DORA metrics for measuring DevOps performance.

CO2: Apply Git branching strategies and implement Continuous Integration pipelines using Jenkins and GitHub Actions.

CO3: Design and deploy containerized applications using Docker and orchestrate workloads using Kubernetes and Helm.

CO4: Implement Infrastructure as Code using Terraform and Ansible and build end-to-end CI/CD pipelines with GitOps.

CO5: Configure monitoring/observability stacks (Prometheus, Grafana, ELK) and integrate security practices in DevSecOps pipelines.

Course Contents

Unit	Description	Duration [Hrs]
I	**DevOps Foundations and Agile Practices** Evolution from Waterfall to Agile to DevOps; CAMS Model; DevOps Lifecycle (Plan-Code-Build-Test-Release-Deploy-Operate-Monitor); Agile/Scrum; Kanban; Lean and Value Stream Mapping; DORA Metrics; Introduction to Site Reliability Engineering (SRE).	9
II	**Version Control and Continuous Integration** Git Internals; Branching Strategies: Git Flow, Trunk-Based Development; Merge/Rebase; GitHub/GitLab PRs and Code Review;	9

	Jenkins Architecture and Declarative Pipelines; GitHub Actions: Workflows, Events, Runners; SonarQube Static Analysis; JFrog Artifactory. [Advanced Topics]: Trunk-Based Development at Scale: feature flags and branch by abstraction; Software Supply Chain Security: SLSA framework; Sigstore (Cosign, Fulcio, Rekor) for artifact signing; Software Bill of Materials (SBOM): CycloneDX and SPDX; Tekton Pipelines for cloud-native CI/CD; Dagger: portable CI/CD pipelines as code using containerised functions; AI-assisted Code Review with LLM-integrated CI workflows; Pipeline Optimisation: caching, parallelism, and test impact analysis.	
III	**Containerisation and Container Orchestration** Docker Architecture; Dockerfile Best Practices; Multi-Stage Builds; Docker Compose; Container Networking and Volumes; Kubernetes Architecture: Control Plane, Worker Nodes, etcd; Pods, Deployments, Services, ConfigMaps, Secrets; Helm Charts; Deployment Strategies: Rolling Update, Blue-Green, Canary.	**9**
IV	**Infrastructure as Code and Continuous Delivery** IaC Concepts; Terraform: Providers, Resources, Variables, Modules, State Management; Terraform Workflow; Ansible: Inventory, Playbooks, Roles, Modules; Configuration Drift and Idempotency; Argo CD and GitOps Principles; End-to-End CI/CD Pipeline; Feature Flags; Environment Promotion. [Advanced Topics]: Crossplane: Kubernetes-native IaC for multi-cloud provisioning; Pulumi: IaC with Python, TypeScript, Go vs. HCL; Policy as Code with OPA and Sentinel; Multi-Cluster GitOps with Fleet and Argo CD; Ephemeral Environments: preview environments per pull request; Progressive Delivery: Flagger for canary analysis; FinOps with IaC: resource tagging and cost estimation (Infracost); Immutable Infrastructure: AMI baking with Packer and Vault dynamic secrets.	**9**
V	**Monitoring, Observability, DevSecOps and Emerging Trends** Observability Pillars: Metrics, Logs, Traces; Prometheus and PromQL; Grafana Dashboards; ELK Stack; Distributed Tracing (Jaeger); SLOs, SLAs, SLIs, Error Budgets; Incident Management; DevSecOps: SAST, DAST; Container Image Scanning: Trivy, Snyk; HashiCorp Vault; AI/MLOps Integration; Green DevOps.	**9**
	Total	**45**

Text Books:

1. Gene Kim, Jez Humble, Patrick Debois, John Willis, The DevOps Handbook, IT Revolution Press, 2nd Ed. (2021).
2. Jez Humble and David Farley, Continuous Delivery, Addison-Wesley, 1st Ed. (2010).
3. Kief Morris, Infrastructure as Code: Dynamic Systems for the Cloud Age, O'Reilly Media, 2nd Ed. (2020)

Reference Books:

1. Brendan Burns et al., Kubernetes: Up and Running, O'Reilly Media, 3rd Ed. (2022).
2. Seth Vargo, James Turnbull, Terraform: Up and Running, O'Reilly Media, 3rd Ed. (2022).
3. Gene Kim et al., The Phoenix Project, IT Revolution Press, 3rd Ed. (2018).

MOOC / NPTEL / e-sources:

1. IBM DevOps and Software Engineering Professional Certificate – Coursera: https://www.coursera.org
2. Linux Foundation LFS261 – DevOps and SRE Fundamentals: https://training.linuxfoundation.org
3. NPTEL – DevOps: https://swayam.gov.in

Program	F.Y. M.Tech (Computer Engineering)			Semester: II			
Course	Block chain Technology			Code:		MCE26PEC-555B	
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
4	4	-	-	25	25	50	100

Pre-requisites: Prior knowledge Foundations of Cryptography, Computer Networks is required.

Course Objectives: This course aims at enabling students:

1. To understand the fundamental concepts of block chain and distributed ledger technologies.
2. To study cryptographic principles used in block chain systems.
3. To analyze consensus mechanisms and security aspects of block chain networks.
4. To design and implement smart contracts and decentralized applications.
5. To explore real-world applications and emerging trends in block chain technology.

Course Outcome: After completion of the course, the students will be able to:

CO1: Explain the architecture and working principles of blockchain systems.
CO2: Apply cryptographic techniques such as hashing and digital signatures in blockchain applications.
CO3: Analyze various consensus mechanisms and evaluate their performance and security.
CO4: Develop and deploy smart contracts and decentralized applications.
CO5: Evaluate real-world blockchain use cases and identify research challenges.

Course Contents

Unit	Description	Duration [Hrs]
I	**Introduction to Blockchain and Distributed Systems** Introduction to distributed systems and peer-to-peer networks. Evolution of blockchain technology. Structure of a block and blockchain architecture. Types of blockchain: public, private and consortium. Bitcoin architecture and transaction lifecycle. Merkle trees and blockchain data structures. Limitations of traditional centralized systems and motivation for blockchain.	12
II	**Cryptographic Foundations of Blockchain** Cryptographic hash functions and properties. Digital signatures and public key cryptography. Elliptic Curve Cryptography (conceptual understanding). Hash pointers and Merkle trees. Proof of Work mechanism. Security principles in blockchain. Attacks on blockchain including double spending and 51% attack.	12
III	**Consensus Mechanisms and Blockchain Platforms** Consensus in distributed systems. Proof of Work, Proof of Stake, Delegated Proof of Stake and Practical Byzantine Fault Tolerance. Comparison of consensus algorithms. Ethereum architecture and	12

	transaction model. Smart contracts and gas mechanism. Introduction to Hyperledger Fabric architecture.	
IV	**Smart Contracts and Decentralized Applications** Smart contract concepts and lifecycle. Solidity programming basics. Development and deployment of smart contracts. Decentralized Applications (DApps) architecture. Web3 and interaction with blockchain. Testing and security issues in smart contracts. Case studies in finance, supply chain and healthcare.	**12**
V	**Advanced Topics and Applications of Blockchain** Scalability issues and solutions such as sharding and Layer-2 solutions. Blockchain interoperability. Blockchain in IoT and Industry 4.0. Decentralized Finance (DeFi) overview. Non-Fungible Tokens (NFTs). Regulatory and legal aspects of blockchain. Research challenges and future trends.	**12**
	Total	**60**

Text Books:

1. Swan, M., Blockchain: Blueprint for a New Economy, O'Reilly Media, 2015. ISBN: 978-1491920497.
2. Antonopoulos, A. M., Mastering Bitcoin: Unlocking Digital Cryptocurrencies, 2nd Edition, O'Reilly Media, 2017. ISBN: 978-1491954386.

Reference Books:

1. Bahga, A., and Madisetti, V., Blockchain Applications: A Hands-On Approach, VPT Publications, 2017.
2. Bashir, I., Mastering Blockchain, 3rd Edition, Packt Publishing, 2020. ISBN: 978-1839213199.
3. Mougayar, W., The Business Blockchain, Wiley, 2016. ISBN: 978-1119300311.

MOOC / NPTEL/YouTube Links:

1. Blockchain Architecture Design and Use Cases, IIT Kharagpur, SWAYAM-NPTEL Course, Available at: https://onlinecourses.nptel.ac.in/noc21_cs53/preview
2. Blockchain and its Applications, IIT Bombay, NPTEL Course, Available at: https://nptel.ac.in/courses/106/105/106105184/
3. Blockchain Technology and Applications, IIT Delhi, SWAYAM-NPTEL Course, Available at: https://onlinecourses.nptel.ac.in/noc22_cs44/preview

Program	F.Y. M.Tech (Computer Engineering)			Semester: II			
Course	Generative Artificial Intelligence			Code:	MCE26PEC-555C		
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Machine Learning, deep learning frameworks is required..

Course Objectives: This course aims at enabling Students

1. To understand advanced concepts of generative modeling and deep learning architectures
2. To analyze and implement modern generative AI models including GANs, VAEs, Transformers, and Diffusion models.
3. To evaluate real-world applications, ethical implications, and deployment challenges of generative AI systems.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Analyze mathematical foundations of generative models and deep learning frameworks.
CO2: Design and implement advanced generative models for text, image, and multimodal data.
CO3: Critically evaluate generative AI systems using appropriate metrics and validation strategies.
CO4: Apply generative AI techniques in research and industry-oriented problem solving.

Course Contents

Unit	Description	Duration [Hrs]
I	**Foundations of Generative Modeling** Probabilistic modeling and latent variable models, generative vs discriminative approaches, maximum likelihood estimation, Bayesian learning basics, neural network foundations, representation learning, optimization challenges in deep generative models.	9
II	**GANs and Variational Autoencoders** Autoencoders and latent space learning, variational inference and ELBO, GAN framework and min-max optimization, conditional GANs, Wasserstein GAN, StyleGAN overview, training instability issues, evaluation metrics such as FID and IS	9
III	**Transformers, Diffusion Models, and Applications** Self-attention and transformer architecture, large language models and pre-training strategies, prompt engineering and fine-tuning methods, diffusion probabilistic models and denoising processes, multimodal generative models, ethical considerations, bias mitigation, intellectual property issues, deployment and scalability considerations.	9
IV	**Diffusion Models and Multimodal Generative Systems** Introduction to diffusion models, forward and reverse diffusion process, basic denoising diffusion probabilistic models, comparison of GANs and diffusion models, text-to-image generation concepts, multimodal	9

	learning basics, vision-language models overview, evaluation metrics such as FID and BLEU, computational considerations and scalability challenges.	
V	**Advanced Applications, Ethics and Deployment of Generative AI** Applications of generative AI in healthcare, finance and content creation, AI safety and alignment concepts, bias and fairness issues in generative models, intellectual property and copyright concerns, responsible AI principles, basics of model deployment and cloud-based integration, overview of MLOps practices, recent industry case studies and emerging trends.	**9**
	Total	**45**

Text Books:

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning", MIT Press, 2016, ISBN: 9780262035613.
2. David Foster, "Generative Deep Learning", 2nd Edition, O'Reilly Media, 2022, ISBN: 9781098134181.
3. Sebastian Raschka, "Build a Large Language Model (From Scratch)", Manning Publications, 2024, ISBN: 9781633437166.

Reference Books:

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning", MIT Press, 2016, ISBN: 9780262035613.
2. David Foster, "Generative Deep Learning", 2nd Edition, O'Reilly Media, 2022, ISBN: 9781098134181.
3. Sebastian Raschka, "Build a Large Language Model (From Scratch)", Manning Publications, 2024, ISBN: 9781633437166.

e-Books:

1. https://dokumen.pub/generative-deep-learning-teaching-machines-to-paint-write-compose-and-play-2nbsped-1098134184-9781098134181.html

MOOC / NPTEL / e-sources:

1. Programming with Generative AI (IISc + NPTEL) – Video syllabus and lectures from the NPTEL course "Programming with Generative AI" (multiple videos online). https://archive.nptel.ac.in/content/syllabus_pdf/106108703.pdf
2. NPTEL Video Lectures: Generative AI Concepts – Part of the Artificial Intelligence: Concepts and Techniques course that includes a lecture on Generative AI. https://onlinecourses.nptel.ac.in/noc25_cs159/preview (Course with Generative AI topic)

Program	M.Tech (All Programs)			Semester : II			
Course	Strategic Management and Business Analytics			Code:	MCE26OEC- 556A		
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Fundamentals of Management Principles is required.

Course Objectives: This course aims at enabling Students

1. To understand strategic management concepts, competitive positioning, and technology-driven business transformation.
2. To apply analytical frameworks and quantitative techniques for strategic decision-making.
3. To use business analytics tools for data-driven strategy formulation and performance evaluation.
4. To analyze digital business models, innovation ecosystems, and platform-based strategies.
5. To evaluate strategic risks, governance, and ethical implications of analytics-driven enterprises.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Analyze strategic intent, stakeholder roles, and business performance using CSFs, KPIs, and KRAs supported by business analytics.

CO2: Evaluate external and internal environments using ETOP, Porter's Five Forces, VRIO, and Value Chain frameworks integrated with analytical modeling.

CO3: Develop strategic plans using portfolio models (BCG, GE), forecasting techniques, predictive analytics, and data-driven decision tools.

CO4: Apply strategic implementation frameworks (Mintzberg's 5Ps, McKinsey 7S, Six Sigma) and HR analytics to monitor and improve organizational performance.

CO5:Design sustainable, data-driven business strategies incorporating Balanced Scorecard, digital business models, AI-driven analytics, and ethical considerations.

Course Contents

Unit	Description	Duration [Hrs]
I	**Foundations of Strategic Management** Concept of strategy, Corporate, Business and Functional Levels of Strategy. Strategic Management Process- Phases; Stakeholder in Business and their Roles in Strategic management. Hierarchy of Strategic Intent; Business Definition, objectives and goals, Linking objectives to mission & vision. Critical success factors (CSF), Key Performance	09

	Indicators (KPI), Key Result Areas (KRA). Analytics: Business Analytics for KPIs such as People, Operations, Staregy e.g. Sales, Profit, ROI, operational efficiency, Revenue , total cost , employee turnover, capacity utilization, defect rate, on time delivery, time to market.	
II	**Formulating Strategy** Analysing External Environment: Environmental Analysis (ETOP), Industry Analysis - Porter's Five Forces Model of competition, Entry & Exit Barriers, Strategic Group analysis. Analytics for Poter's five forces: Case or Cases which include - Statistical modeling to predict the likelihood of new entrants based on various factors.Analysing Company's Environment:Resource-Based View- Examine a firm's resources and competitive position, introducing the VRIO Framework and benchmarking, Porter's Value Chain Analysis	**09**
III	**Strategic Planning: Portfolio Analysis** Business Portfolio Analysis - BCG Matrix – GE 9 Cell Model, Generic Competitive and Grand Strategies: Analytics: Market trends, competitor strategies, and emerging opportunities, Strategic Insights for Managers, Forecasting, Business Simulations, Uncertainty estimations, Use Cases involving Predictive tools and Applications in solving problems using Marketing, Finance, M & A, Operations and Supply chain analytics, Competitive advantage through data-driven decisions. Cases and Applications in different sectors	**09**
IV	**Strategic Implementation and Evaluation** Structural, Functional Behavioural Implementation, Strategy, Mintzberg's 5 Ps – Deliberate & Emergent Strategies , Mc Kinsey's 7s Framework, Benchmarking, Six Sigma ,Analytics: Employee engagement and performance measurement, Case studies: Enhancing HR practices with analytics.	**09**
V	**Introduction to Business Analytics** Descriptive, Predictive, Prescriptive analytics,Data-driven decision-making,Data Warehousing and ETL concepts, Statistical tools for analytics:Regression,Time series analysis,Hypothesis testing,Introduction to Machine Learning for business applications	**09**
	Total	**45**

Text Books:

1. Strategic Management: Concepts and Cases-Fred R. David & Forest R. David

Reference Books:

1. The analytics revolution: how to improve your business by making analytics operational in the big data era, Bill Franks. Hoboken: Wiley

2. Statistical data analysis explained: applied environmental statistics with R, Clemens Reimann. Chichester: John Wiley and Sons
3. Data Analytics using R, Seema Acharya, TMGH

e-sources:

1. https://www.youtube.com/watch?v=lzOFcVurWAk

e-Books:

1. https://www.phindia.com/Books/ShowBooks/MjY2/Business-Analysis-Analytics?srsltid=AfmBOoq96J6fynl5WHLK_fDUwl-HwFtXYXOOcbkID4F5EN9uxGmQmP_d

MOOC / NPTEL/YouTube Links:

1. https://www.youtube.com/watch?v=67ZLG42Iwfk&list=PL3sk8amcDFDnzznNGtE0-BBt-fBgOkGVM
2. https://nptel.ac.in/courses/110107509
3. https://onlinecourses.nptel.ac.in/noc26_cs64/preview

Program	M.Tech (All Programs)			Semester : II			
Course	Software Project Management			Code:	MCE26OEC-556B		
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: Prior knowledge of Software Engineering, Basics of Project planning and management is required.

Course Objectives: This course aims at enabling Students

1. To understand core principles and practices of managing software projects.
2. To plan, estimate, schedule and control software development activities.
3. To identify and mitigate risks systematically.
4. To apply modern methods (Agile, iterative) along with classic planning techniques.
5. To analyze team, communication and stakeholder challenges in real work scenarios.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Understand foundational concepts and challenges in software project management.
CO2: Prepare project plans including estimation, schedules and budgets.
CO3: Apply risk, quality and resource management techniques to projects.
CO4: Use project control, change and configuration management frameworks.
CO5: Lead and communicate effectively in multidisciplinary software teams.

Course Contents

Unit	Description	Duration [Hrs]
I	**Project Concepts and Management** Project life cycle models-ISO 9001 model-Capability Maturity Model-Project Planning-Project tracking-Project closure. Evolution of Software Economics – Software Management Process Framework: Phases, Artifacts, Workflows, Checkpoints – Software Management Disciplines: Planning / Project Organization and Responsibilities / Automation / Project Control – Modern Project Profiles	09
II	**Cost Estimation and Agile Development** Problems in Software Estimation – Algorithmic Cost Estimation Process, Function Points, SLIM (Software Life cycle Management), COCOMO II (Constructive Cost Model) – Estimating Web Application Development – Concepts of Finance, Activity Based Costing and Economic Value Added (EVA) – Balanced Score Card. Agile methods, Agile development techniques, Extreme Programming, Various Agile Process Models – ASD, SCRUM, DSDM, Crystal, FDD, LSD, AM, AUP.	09

III	**Software Quality Management** Software Quality Factors – Software Quality Components – Software Quality Plan – Software Quality Metrics – Software Quality Costs – Software Quality Assurance Standard – Certification – Assessment. In Stream Activities in Project Management: Software Measurement Framework, Ishikawa's Seven tools, Process Assessment and patterns, CMMI – IPPD, Product and Process attributes, Software Quality and configuration management	**09**
IV	**Software Management and Metrics** Software Configuration Management – Risk Management: Risk Assessment: Identification / Analysis / Prioritization – Risk Control: Planning / Resolution / Monitoring – Failure Mode and Effects Analysis (FMEA) – Defect Management – Cost Management. Software Metrics – Classification of Software Metrics: Product Metrics: Size Metrics, Complexity Metrics, Halstead's Product Metrics, Quality Metrics, and Process metrics.	**09**
V	**Project Evaluation and Emerging Trends** Strategic Assessment–Technical Assessment–Cost Benefit Analysis–Cash Flow Forecasting–Cost Benefit Evaluation Technique–Risk Evaluation–Software Effort Estimation. Emerging Trends in project management	**09**
	Total	**45**

Text Books:

1. Roger S. Pressman, Software Engineering: A practitioners approach, TMH , Seventh Edition, ISBN 978–0–07–337597–7 , ISBN 0–07–337597–7.
2. Ian Sommerville, Software Engineering, Addison-Wesley, Tenth Ed. ISBN-13: 978-0133943030 ISBN-10: 0133943038

Reference Books:

1. Linda I. Shafer, Robert T. Futrell, Donald F. Shafer, Quality Software Project Management, Prentice Hall, ISBN 0130912972.
2. Scott Berkun, The Art of Project Management, O'Reilly, First Edition, ISBN 0596007868.
3. Orit Hazzan and Yael Dubinsky, Agile software engineering, Springer –Verlag London, First Edition, ISBN 978-1-84800-199-2
4. Pankaj Jalote, Software Project Management in practice, Addison-Wesley Professional, ISBN 0201737213.
5. Craig Larman, Applying UML and Patterns, Pearson Education, Third Edition.
6. Grady Booch, James Rambaugh, Ivar Jacobson, Unified Modeling Language Users Guide, Addison-Wesley, Second Edition, ISBN 0321267974.

e-sources:

1. https://books.google.mw/books?id=JUwQz2A_k_gC&printsec=frontcover#v=onepage&q&f=false

e-Books:

1. https://www.amazon.in/Software-Project-Management-Chandramouli-Dutt-ebook/dp/B07YG4S9D9

MOOC / NPTEL/YouTube Links:

1. https://www.youtube.com/playlist?list=PLI0WdhayQoFC3JUXMzakqmst7l9W0Ljw
2. https://nptel.ac.in/courses/106105218

Program	M.Tech (All Programs)			Semester : II			
Course	Cost management of engineering projects			Code:	MCE26OEC- 556C		
Credits	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	UT	FA	SA	Total
3	3	-	-	25	25	50	100

Pre-requisites: None

Course Objectives: This course aims at enabling Students

6. To understand various cost concepts and their role in managerial decision-making.
7. To analyze project management processes including planning, execution, and control.
8. To apply cost behavior techniques and profit planning tools in real-world scenarios.
9. To evaluate modern cost management practices such as ABC, benchmarking, and budgeting.
10. To utilize quantitative techniques for optimizing cost and improving project efficiency.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Apply cost concepts and costing systems for effective managerial decision-making.

CO2: Analyze and manage project execution, including cost estimation and control techniques.

CO3: Evaluate cost behavior and use profit planning tools for business decisions.

CO4: Implement advanced cost management methods like ABC, budgeting, and performance evaluation.

CO5: Apply quantitative techniques such as LP, PERT/CPM, and simulation for optimizing project costs.

Course Contents

Unit	Description	Duration [Hrs]
I	**Fundamentals of Strategic Cost Management** Introduction and Overview of the Strategic Cost Management Process Cost concepts in decision making; Relevant cost, Differential cost, Incremental cost and Opportunity cost. Objectives of a Costing System; Inventory valuation; Creation of a Database for operational control; Provision of data for Decision-Making.	09
II	**Project Management and Cost Control** Project: meaning, Different types, why to manage, cost overruns centers, various stages of project execution: conception to commissioning. Project execution as conglomeration of technical and nontechnical activities. Detailed Engineering activities. Pre-project execution main clearances and documents Project team: Role of each member. Importance Project	09

	site: Data required with significance. Project contracts. Types and contents. Project execution Project cost control. Bar charts and Network diagram. Project commissioning: mechanical and process	
III	**Cost Behavior and Profit Planning** Cost Behavior and Profit Planning Marginal Costing; Distinction between Marginal Costing and Absorption Costing; Break-even Analysis, Cost-Volume-Profit Analysis. Various decision-making problems. Standard Costing and Variance Analysis. Pricing strategies: Pareto Analysis. Target costing, Life Cycle Costing. Costing of service sector. Just-in-time approach, Material Requirement Planning, Enterprise Resource Planning, Total Quality Management and Theory of constraints	**09**
IV	**Advanced Cost Management Techniques** Activity-Based Cost Management, Bench Marking; Balanced Score Card and Value-Chain Analysis. Budgetary Control; Flexible Budgets; Performance budgets; Zero-based budgets. Measurement of Divisional profitability pricing decisions including transfer pricing	**09**
V	**Quantitative Techniques for Cost Optimization** Quantitative techniques for cost management, Linear Programming, PERT/CPM, Transportation problems, Assignment problems, Simulation, Learning Curve Theory.	**09**
	Total	**45**

Text Books:

4. Cost Accounting A Managerial Emphasis, Prentice Hall of India, New Delhi
5. Charles T. Horngren and George Foster, Advanced Management Accounting
6. Robert S Kaplan Anthony A. Alkinson, Management & Cost Accounting

Reference Books:

1. Ashish K. Bhattacharya, Principles & Practices of Cost Accounting A. H. Wheeler publisher
2. N.D. Vohra, Quantitative Techniques in Management, Tata McGraw Hill Book Co. Ltd

MOOC / NPTEL/YouTube Links:

4. NPTEL – Management Accounting : https://nptel.ac.in/courses/110107127
5. NPTEL – Strategic Sourcing / Pricing Strategy: https://onlinecourses.nptel.ac.in/noc24_mg57/preview

Program	F.Y. M.Tech (Computer Engineering)			Semester: II			
Course	Research Seminar-I			Code:	MCE26ELC-557		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
2	-	4	-	25	25	-	50

Pre-requisites: Prior knowledge of soft skill is required.

Course Objectives: This course aims at enabling students:

1. To deepen technical knowledge by exploring specialized topics beyond the regular curriculum.
2. To develop research and analytical skills through literature survey, information evaluation, and technical synthesis.
3. To enhance communication and presentation skills for effective delivery of technical content.
4. To promote critical thinking and independent learning in preparation for thesis/dissertation and professional careers.

Course Outcomes: After completion of the course, the students will be able to:

CO1: formulate research goals and objectives in a chosen technical domain.

CO2: Search and synthesize technical literature from various sources.

CO3: Analyse and interpret technical data and construct logical reasoning.

CO4: Prepare and present technical documents and seminars as per academic and professional standards.

Guidelines for Students:

1. Individual Responsibility: The seminar must be carried out individually. Students must regularly meet their assigned guide (at least once a week), maintain a progress diary with remarks, and adhere to timelines and deadlines.
2. Topic Selection & Approval: Select a domain-specific topic relevant to Computer Engineering (e.g., AI/ML, Cybersecurity, IoT, Cloud, Data Science, etc.) focusing on recent trends, emerging technologies, or advanced concepts. The topic must have appropriate scope and be approved by the faculty supervisor.
3. Relevance & Novelty: The topic should demonstrate depth, current advancements, open challenges, or future research directions. Avoid purely introductory or textbook-based topics.
4. Ethical Research Practice: Follow academic integrity standards, conduct ethical research, properly cite all references, and strictly avoid plagiarism.
5. Topic Proposal Submission: Submit a 1–2 page proposal including title, abstract, motivation, relevance, tentative outline, and at least 5–7 quality references within 2–3 weeks of topic approval.

6. Literature Review & Critical Analysis: Use peer-reviewed sources (IEEE, ACM, Springer, arXiv, etc.). Go beyond summarization—critically analyze methodologies, compare approaches, identify limitations, and suggest future directions.
7. Seminar Report Preparation: Submit a 20–25 page report (IEEE format) including Abstract, Introduction, Literature Review, Core Content, Analysis/Discussion, Future Scope/Conclusion, and References. The report must be approved and signed by the guide before submission.
8. Oral Presentation: Deliver a 25–30 minute presentation followed by Q&A. Use high-quality slides, ensure clear articulation, confident delivery, and demonstrate thorough understanding of the topic.
9. Incorporation of Feedback: Actively incorporate suggestions from the guide and reviewers to improve the technical depth, clarity, and rigor of the seminar work.

Course Syllabus

Semester-III

Program	S.Y. M.Tech(Computer Engineering)			Semester: III			
Course	Massive Open Online Course			Code:	MCE26SL-601		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
4	-	-	-	100	-	-	100

Course Objectives: This course aims at enabling students:

1. To encourage independent learning through reputed MOOC platforms.
2. To develop expertise in emerging areas of Computer Engineering.
3. To enhance professional skills aligned with industry requirements.
4. To promote self-paced technical competency development.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Analyze advanced concepts in the selected domain
CO2: Apply techniques to solve practical problems
CO3: Demonstrate independent learning and certification competency
CO4: Present technical knowledge effectively through seminar and viva

Guidelines for Students:

1. Individual students need to register for MOOC course of their interest. Course must be **approved by Project Guide before registration.**
2. Week assignments need to be regularly completed as per requirement of course and to be submitted in file to Project Guide, which will be considered for internal assessment of course.
3. The certification of course is mandatory.
4. Term work marks will be awarded to the students by reviewer panel consist of Project guide and senior faculty members. Review will be conducted based on the rubrics which will include the parameters such as Oral, Presentation, assignments related to the course etc.

Program	S.Y. M.Tech (Computer Engineering)			Semester: III			
Course	Internship			Code:	MCE26ELC - 602		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	PR	OR	TW	Total
5	-	10	-	-	-	100	100

Pre-requisites: Prior Knowledge of object-oriented concepts, data structures, and (C/C++/Java/Python) programming is required.

Course Objectives: This course aims at enabling students.

1. To learn and understand real life/industrial situations.
2. To get familiar with various tools and technologies used in industries and their applications.
3. To nurture professional and societal ethics.
4. To create awareness of social, economic and administrative considerations in the working environment of industry organizations.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Demonstrate professional competence through industry internship.

CO2: Apply knowledge gained through internships to complete academic activities in a professional manner.

CO3: Choose appropriate technology and tools to solve given problem.

CO4: Demonstrate abilities of a responsible professional and use ethical practices in day to day life.

CO5: Creating network and social circle, and developing relationships with industry people.

CO6: Analyze various career opportunities and decide carrier goals.

Course Contents

Preamble:

The Institute will allow internships in various organizations. Students will be allotted internship in a company based on merit or any other criterion laid down by the company. In some cases, the students may be allowed to arrange internship. In that case a student can take a letter from the placement/respective department and contact the company. The confirmation letter from the company has to be submitted in the placement / respective department.

Students will be associated with one faculty from respective department who will act as internal mentor. After internship duration, internal mentor will assess the student's performance.

Students will maintain the record of the work done in the industry and submit a report in the institute within one week after completion of internship. The certificate has to be duly signed by an official of the company. The evaluation date will be notified at least one week before by internship coordinator.

General Guidelines:

Student can take internship work in the form of Online/Onsite mode from any of the following but not limited to:

- Industry / Government Organization Internship
- Eduskills, Internshala, ByteXL (Platinum Placement) etc.
- EDC Cell and startups cells of institute / In-house product development, intercollegiate, inter department research internship under research lab/group etc.
    - Here, students can work as an Incubitee under Incubation and Innovation Cell at Institute (CIIL) and should get registered as a Start-up to avail the Internship opportunity.
- Research internship under Professors (Internal and External), IISC, IIT's, NIT's and other Research organizations.
- Participate in open-source contribution

Any other with the permission of faculty mentor.

Internship Progress Monitoring:

Internship Process Flow:

Internship opportunity will be provided by 3 ways- Through Training & Placement Cell, Through Department/Faculty and searched by own [students].

1. Internship opportunity will be provided by 3 ways- Through Training & Placement Cell,Through Department/Faculty/Searched by own (Students).
2. The student has to take permission from faculty mentor and internship coordinator to verify quality of internship.
3. Student has to submit the Internship offer letter or official mail communication proof to internship coordinator.
4. After permission the student needs to start the Internship program.
5. The intern student must report about the Internship program to faculty mentor.

6. At the end of the Internship program, student should submit completion certificate from industry and report.
7. Student feedback of Internship needs to be submitted by intern student.

Internship Work Evaluation:

- Students will be allotted to faculty mentors.
- Students will be allowed to do internship at the end of 2nd semester and till the commencement of 3rd semester.
- Students will do internship for minimum of 4-weeks to 6-weeks (40-45 hours/week) through which they can earn 5 credits.
- If students are in summer term, then they have to manage their time for summer term academic activities (if any) and internship.

Annexure:

Following are the Internship forms and letter required to be submitted at the department, as per the Internship progression:

- Form 1: Student Internship Program Application as permission letter
- Form 2: Student feedback of Internship
- Final report with completion certificate
- Rubrics for internship work evaluation

Performance Indicators [PI]

Sr.No	Student's Name	I: Technical Knowledge [30Marks]	II: Presentation Skills [20 Marks]	III: Outcomes [30Mark]	IV: Professional Behavior [10 Marks]	V: Quality of Report [10 Marks]	Total Marks [100Marks]

Program	S.Y. M.Tech (Computer Engineering)			Semester: III			
Course	Research Seminar-II			Code:	MCE26ELC - 603		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
4	-	8	-	25	25	-	50

Pre-requisites: Prior knowledge of soft skill & research seminar-II is required.

Course Objectives: This course aims at enabling students:

1. To develop research aptitude by guiding students in selecting, analyzing, and defining a focused research problem in emerging areas.
2. To build competency in literature review and patent analysis using credible scientific sources while identifying research gaps.
3. To inculcate ethical research practices and intellectual property awareness in accordance with academic standards.
4. To enhance technical writing, documentation, and **Pre-requisites** sentation skills required for successful execution of report.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Identify and formulate a clear research problem with defined objectives, scope, and methodology

CO2: Critically analyze research literature and patents to identify gaps and justify the proposed work.

CO3: Demonstrate ethical research practices by maintaining academic integrity and proper citation standards.

CO4: Prepare and present a structured technical seminar report and effectively defend their research work..

Guidelines for Students:

The Research Project Seminar is the initial phase of the master's thesis work. It helps students build knowledge in intellectual property rights and research ethics while conducting patent searches and reviewing related literature to understand the current status of the chosen research area.

Its primary objective is to prepare students for the Final Project. This includes identifying or developing suitable methods and tools, defining project requirements, and planning experiments or tests to validate the proposed solution in line with recent research trends.

1. Individual Project-Based Work: Each student must independently carry out the seminar based on their approved research project topic in consultation with the assigned guide.
2. Topic Selection & Approval: Select a relevant, focused, and trend-based topic in Computer Engineering (e.g., AI/ML, Cybersecurity, IoT, Cloud, etc.), ensuring novelty and depth, and obtain prior approval from the faculty supervisor.

3. Research Planning & Objectives: Formulate clear research goals, objectives, and scope; plan pilot studies or experiments considering professional domain requirements.
4. Literature Survey & Critical Analysis: Conduct an in-depth review using credible sources (IEEE, ACM, Springer, arXiv, etc.), analyze different approaches, compare methodologies, identify research gaps, and build evidence-based conclusions.
5. Ethics & Academic Integrity: Follow ethical research practices, avoid plagiarism, properly cite references, and maintain academic honesty throughout the work.
6. Regular Progress & Documentation: Meet the guide at least once a week, follow deadlines, incorporate feedback, and maintain a project diary recording meetings, remarks, and progress for review presentations.
7. Seminar Report Preparation: Prepare a well-structured technical report (approx. 20–25 pages) in standard academic format (IEEE style) including Abstract, Introduction, Literature Review, Core Content, Analysis, Conclusion/Future Scope, and References.
8. Scientific Writing Skills: Write in a formal scientific/technical style using appropriate terminology; design and edit documents as per institutional/publisher guidelines.
9. Oral Presentation & Defense: Deliver a 45-60 minute structured presentation using high-quality visual aids, demonstrate clarity and confidence, and effectively answer questions during the Q&A session.
10. Evaluation Criteria Awareness: Performance will be assessed based on topic relevance, research depth, report quality, originality, presentation skills, and ability to defend the work.

Program	S.Y. M.Tech (Computer Engineering)			Semester: III			
Course	Research Project – I			Code:	MCE26ELC - 604		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
9	-	18	-	25	25	-	50

Pre-requisites: Prior knowledge of software project management is required.

Course Objectives: This course aims at enabling students:

1. To enable students to identify and define a real-world research problem relevant to their discipline.
2. To develop the ability to design and plan a systematic research methodology.
3. To encourage independent research, critical thinking, and innovative solution development.
4. To train students in professional documentation, ethical research practices, and dissertation writing.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Analyse and formulate a contemporary engineering research problem.

CO2: Implement appropriate research methodologies, prototypes, simulations, or software solutions.

CO3: Demonstrate independent research capability, critical analysis, and regular progress documentation.

CO4: Prepare and present a high-quality dissertation following proper citation standards and academic integrity.

Guidelines for Students:

The master's program concludes with a research project independently designed by the student. This project is presented in the form of a final research report or dissertation under the guidance of an academic supervisor. Students are expected to identify a relevant and contemporary real-world problem and apply the engineering principles they have learned to develop appropriate solutions. These solutions may involve designing prototypes, conducting simulations, developing software, or establishing processes, syntheses, or correlations. The research project should demonstrate originality, carefully justified methodological decisions, and meaningful contribution to ongoing discussions within the discipline. The dissertation must reflect the highest standard of research, critical thinking, and analytical ability that the student can achieve.

1. The dissertation is a year-long project conducted in two phases and may be carried out either in-house or in collaboration with industry, as assigned by the department.
2. At the beginning of Phase-I, students are allotted a project topic and internal guide, who supports project planning and evaluation.

3. Phase I includes essential research components such as literature survey, problem definition, motivation and objectives, preliminary design/feasibility analysis, and overall research design.
4. Students are encouraged to initiate their research through informal discussions with guides, peers, and research professionals to explore and refine potential project ideas.
5. The first formal milestone is to identify a provisional project title and prepare an abstract (up to 500 words), demonstrating clarity of research intent and proposal-writing skills.
6. Students must prepare a detailed project proposal, understanding that research topics may evolve; regular communication with the guide is expected for necessary refinements.
7. During Term-I research, students should dedicate substantial time to their project and actively engage in regular face-to-face or virtual meetings with their supervisor to monitor progress.
8. Maintaining a research notebook (digital or physical) is strongly recommended to document methods, observations, and progress for transparency and academic integrity.
9. Students must ensure secure and regular backup of research data and documents, as data loss will not be accepted as a valid reason for missing deadlines.
10. The final project report must be submitted by the specified deadline, following an appropriate and consistent citation style relevant to the research domain, as advised by the supervisor.

Course Syllabus

Semester-IV

Program	S.Y. M.Tech (Computer Engineering)			Semester: IV			
Course	Research Seminar-III			Code:	MCE26ELC -651		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
4	-	8	-	50	50	-	100

Pre-requisites: Prior knowledge of soft skill & research seminar-II is required.

Course Objectives This course aims at enabling students:

1. To develop research and self-learning skills in emerging areas of Computer Engineering.
2. To enhance critical thinking and analytical abilities through systematic literature review and gap identification.
3. To strengthen technical writing and documentation skills following professional academic standards.
4. To improve presentation and communication skills through structured seminar delivery and discussion.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Identify and select relevant research topics aligned with current trends.

CO2: Conduct ethical research and perform critical literature analysis using credible academic sources.

CO3: Prepare a well-structured technical report in standard academic format with proper citations.

CO4: Effectively present and defend their seminar work, demonstrating subject understanding and professional communication skills.

Guidelines for Students:

The seminar is designed to strengthen students' research, presentation, and critical thinking abilities, equipping them for higher academic studies and professional careers. It offers a supportive platform for students to enhance their self-learning skills through modern information technologies and to effectively apply newly acquired knowledge and competencies in practical and emerging domains.

1. Individual Responsibility: The seminar must be carried out individually. Students should regularly meet their assigned guide (at least once a week), follow timelines, maintain a progress diary with remarks, and incorporate feedback to improve the quality of work.
2. Ethics & Academic Integrity: Students must conduct research ethically, avoid plagiarism, properly cite all references, and adhere to academic standards throughout the seminar work and final report.
3. Topic Selection: The topic must be relevant to Computer Engineering, aligned with current trends or emerging technologies, appropriately scoped, and approved by the

faculty supervisor. Novelty and critical insight are encouraged over simple textbook summaries.

4. Broad Research Areas: Possible domains include Artificial Intelligence, Machine Learning, Data Science, Cybersecurity, Cloud Computing, IoT, Computer Networks, Software Engineering, Embedded Systems, Computer Vision, NLP, Block chain, and Quantum Computing.
5. Topic Proposal Submission: Within 2–3 weeks of approval, students must submit a concise proposal including title, abstract, motivation, preliminary references (5–7 quality sources), and a tentative outline for supervisor approval.
6. Literature Review & Research: Students must rely on peer-reviewed and credible academic sources (IEEE, ACM, Springer, etc.), conduct critical analysis of existing approaches, identify advantages, limitations, and research gaps, and systematically organize notes.
7. Seminar Report Preparation: A detailed report (20–25 pages) should be prepared in professional academic format (e.g., IEEE style), including abstract, introduction, literature review, core content, analysis, future scope, conclusion, and properly formatted references.
8. Oral Presentation: The seminar includes a 45-60 minute presentation followed by Q&A. Students must prepare clear and professional slides, practice effective delivery, and confidently respond to questions demonstrating strong subject understanding.
9. Submission & Approval: The final seminar report must be approved and signed by the guide before submission within the stipulated deadline. All documentation and formatting requirements must be completed properly.

Program	S.Y. M.Tech (Computer Engineering)			Semester: IV			
Course	Research Project – II			Code:	MCE26ELC-652		
Credits	Teaching Scheme (Hrs./Week)			Examination Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
18	-	36	-	150	50	-	200

Pre-requisites: Prior knowledge of software project management is required.

Course Objectives: This course aims at enabling students:

1. To apply the selected research methodology/technology for complete system development or simulation.
2. To analyse experimental results using appropriate datasets, metrics, and comparative studies.
3. To create a structured dissertation and research publication/presentation following academic standards.
4. To demonstrate ethical research practices by maintaining originality, documentation quality, and continuous progress reporting.

Course Outcomes: After completion of the course, the students will be able to:

CO1: Implement the proposed system/model and validate its performance through systematic testing.

CO2: Analyse experimental data using graphs, tables, and comparative performance evaluation techniques.

CO3: Evaluate research findings to draw meaningful conclusions and identify future research scope.

CO4: Create a comprehensive dissertation/report and present or publish the research work adhering to academic integrity standards.

Guidelines for Students:

Research Project II focuses on completing and consolidating the work initiated in Semester III by finalizing the selected technology, carrying out full implementation or simulation, and performing systematic testing, performance evaluation, and comparative analysis to obtain validated results and clear conclusions. The project must include detailed documentation of the experimental setup, datasets, metrics, graphs, and comparisons, along with key contributions, applications, and future scope, and may be strengthened through publication or presentation. Continuous progress should be ensured through regular reviews and weekly meetings with the supervisor, while strictly adhering to research ethics with plagiarism below 15%. The final dissertation (soft and hard copy) must be submitted in the prescribed format, with proper backup of data and code, and will be evaluated based on originality, quality of implementation, analysis, documentation, viva performance, and overall innovative contribution. System Implementation: Ensure full development/simulation and testing of the proposed system or model, maintaining correctness, efficiency, and proper validation of outcomes.

1. Results & Performance Analysis: Present detailed experimental setup, datasets, performance metrics, graphs, tables, and comparisons with existing techniques, clearly highlighting significant findings.
2. Conclusion & Future Scope: Summarize key contributions, applications, and outcomes of the research, and propose potential improvements or extensions for future work.
3. Publication & Validation: Validate findings by publishing or presenting the work in peer-reviewed conferences and/or journals, and attach proof of publication (if available).
4. Progress Monitoring: Demonstrate continuous progress through regular presentations, documentation, and reporting as directed by the PG Coordinator/HOD. Maintain a logbook recording weekly supervisor meetings (minimum once per week) and feedback.
5. Ethics & Academic Integrity: Strictly avoid plagiarism, false results, and unethical practices. Ensure the final plagiarism similarity index does not exceed 15%.
6. Documentation & Report Submission: Prepare a duly certified final dissertation report in the prescribed standard format, including all chapters (implementation, results, conclusion), proper references, and submit both soft copy (PDF) and hard copy as per institute norms.
7. Data & Backup Management: Securely maintain backups of source code, datasets, and reports throughout the research process.